



CENTRO UNIVERSITÁRIO DE BRASÍLIA -UniCEUB
CURSO DE ENGENHARIA DE COMPUTAÇÃO

PAULO UEINER MOREIRA DE ASSIS

**SISTEMA DE RASTREAMENTO DE VEÍCULOS PARA
EMPRESAS DE TRANSPORTE UTILIZANDO
NAVEGAÇÃO POR SATÉLITE**

Orientador: Dr. Miguel Archanjo Bacellar Goes Telles Junior

Brasília

2010

PAULO UEINER MOREIRA DE ASSIS

**SISTEMA DE RASTREAMENTO DE VEÍCULOS PARA
EMPRESAS DE TRANSPORTE, UTILIZANDO
NAVEGAÇÃO POR SATÉLITE**

Trabalho apresentado ao Centro
Universitário de Brasília
(UniCEUB) como pré-requisito
para a obtenção de Certificado de
Conclusão de Curso de Engenharia
de Computação.

Orientador: Dr. Miguel Archanjo.

Brasília

2010

PAULO UEINER MOREIRA DE ASSIS

RA: 2051652/4

**SISTEMA DE RASTREAMENTO DE VEÍCULOS PARA EMPRESAS DE
TRANSPORTE, UTILIZANDO NAVEGAÇÃO POR SATÉLITE**

Trabalho apresentado ao Centro
Universitário de Brasília
(UniCEUB) como pré-requisito
para a obtenção de Certificado de
Conclusão de Curso de Engenharia
de Computação.

Orientador: Dr. Miguel Archanjo

Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro de Computação,
e aprovado em sua forma final pela Faculdade de Tecnologia e Ciências Sociais Aplicadas -
FATECS.

Prof. Abiezer Amarilia Fernandez
Coordenador do Curso

Banca Examinadora:

Prof. Miguel Archanjo Bacellar Goes Telles Junior, Doutor em Geologia.
Orientador - UniCEUB

Prof. Maria Marony Souza Farias Nascimento, Mestre em Engenharia Elétrica.
UniCEUB

Prof. Gil Renato Ribeiro Gonçalves, Doutor em Física.
UniCEUB

Prof. Flávio Antônio Klein, Mestre em Estatística e Métodos Quantitativos.
UniCEUB

Dedico este trabalho ao meu avô Ambrósio Moreira (in memorian). Exemplo de vida. Apesar de sua ausência possui participação viva e marcante em cada página deste trabalho. Descanse em paz meu querido poeta!

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus por tudo de bom que Ele proporciona à minha vida. Depois, não poderia deixar de agradecer meus pais Irene e Juscelino que estiveram ao meu lado em todas as decisões em minha vida. Sem eles eu não teria conseguido tudo que consegui.

Às minhas irmãs Pollyana e Rayana por me ajudarem no que precisei.

Aos amigos de trabalho por estarem sempre comigo, e me ajudarem com conselhos valiosos para que este projeto se tornasse realidade. Em especial ao Calixto Jorge, pelo suporte na biblioteca de códigos FrameCalixto e também ao Marcus Feliciano pela ajuda na biblioteca dos mapas.

Aos amigos Paulo Gabriel e Said Abd que me acompanham desde o início do curso e transformaram esta caminhada de aparência infindável, em um período especial de minha vida, momentos que passaram rapidamente e que sentirei falta. Conquistei dois amigos que serão eternamente lembrados.

Em especial à minha amada namorada Nayane Mota, que em todos os momentos esteve comigo, apoiando e me aconselhando a nunca desistir dos meus ideais.

Estendo os meus agradecimentos ao meu incansável e dedicado professor orientador que muito contribuiu para o desenvolvimento e finalização deste meu projeto. Sob sua tutela meus passos foram guiados. Muito Obrigado Professor Miguel Archanjo!

Meu muito obrigado à banca examinadora pela sua avaliação precisa.

Lembro também, de todos aqueles que de alguma forma contribuíram para que este projeto se concretizasse.

SUMÁRIO

LISTA DE FIGURAS.....	VIII
LISTA DE TABELAS.....	X
LISTA DE SIGLAS E ABREVIATURAS	XI
RESUMO.....	XIII
ABSTRACT	XIV
CAPÍTULO 1 – INTRODUÇÃO	15
1.1 – APRESENTAÇÃO DO PROBLEMA.....	15
1.2 – OBJETIVOS DO TRABALHO.....	16
1.3 – JUSTIFICATIVA E IMPORTÂNCIA DO TRABALHO	16
1.4 – ESCOPO DO TRABALHO.....	17
1.5 – RESULTADOS ESPERADOS.....	18
1.6 – ESTRUTURA DA MONOGRAFIA	18
CAPÍTULO 2 - APRESENTAÇÃO DO PROBLEMA	19
CAPÍTULO 3 – REFERENCIAL TEÓRICO.....	21
3.1 – RASTREAMENTO DE VEÍCULOS (RV).....	21
3.2 – SISTEMA GLOBAL DE NAVEGAÇÃO POR SATÉLITE - GNSS	21
3.2.1 – O SISTEMA GLONASS.....	23
3.2.2 – O SISTEMA GALILEO	24
3.2.3 – SISTEMA DE POSICIONAMENTO GLOBAL - GPS	25
3.2.3.1 – SEGMENTOS GPS	28
3.2.3.2 – FONTES DE ERRO NO SISTEMA GPS	30
3.3 – COORDENADAS GEOGRÁFICAS	31
3.4 – PADRÃO NMEA	33
3.5 – MENSAGEM \$GPRMC	33
3.6 – MÓDULO GPS TRACKER.....	35
3.7 – LINGUAGENS PARA DESENVOLVIMENTO WEB.....	37
3.7.1 – LINGUAGENS DE MARCAÇÃO: HTML, XHTML e DHTML.....	37
3.7.2 – FOLHA DE ESTILO EM CASCATA - CSS	38
3.7.3 – JAVASCRIPT.....	39
3.7.4 – PHP: PREPROCESSADOR DE HIPERTEXTO - VERSÃO 5 (PHP5).....	39
3.8 – PROTOCOLOS E PADRÕES DE COMUNICAÇÃO	40
3.8.1 – SISTEMA GLOBAL PARA COMUNICAÇÕES MÓVEIS - GSM.....	41

3.8.2 – <i>SERVIÇO DE RÁDIO DE PACOTE GERAL - GPRS</i>	42
3.9 – <i>SERVIDORES E BANCO DE DADOS</i>	43
3.9.1 – <i>SERVIDOR WEB APACHE</i>	43
3.9.2 – <i>BANCO DE DADOS POSTGRESQL</i>	44
3.10 – <i>BIBLIOTECAS UTILIZADAS</i>	46
3.10.1 – <i>FRAMEWORK PHP: FRAMECALIXTO</i>	46
3.10.2 – <i>BIBLIOTECA JAVASCRIPT: JQUERY</i>	47
3.10.3 – <i>BIBLIOTECA DE MAPAS SOB DEMANDA: GOOGLE MAPS API</i>	47
CAPÍTULO 4 – MODELO PROPOSTO	49
4.1 – <i>APRESENTAÇÃO GERAL DO MODELO PROPOSTO</i>	49
4.1.1 – <i>ESTIMATIVA DE CUSTOS</i>	51
4.2 – <i>DESCRIÇÃO DAS ETAPAS DO MODELO</i>	52
4.2.1 – <i>INSTALAÇÃO E CONFIGURAÇÃO DO MÓDULO GPS</i>	52
4.2.2 – <i>TRATAMENTO DAS MENSAGENS PROVENIENTES DO MÓDULO GPS</i>	57
4.2.3 – <i>ESTRUTURA DE PASTAS DO SISTEMA</i>	59
4.2.4 – <i>MODELO DE DADOS DO SISTEMA</i>	61
4.2.5 – <i>FUNCIONALIDADES DO SISTEMA</i>	62
4.3 – <i>DESCRIÇÃO DA IMPLEMENTAÇÃO</i>	72
CAPÍTULO 5 – EXPERIMENTO E RESULTADOS	75
5.2 – <i>TESTES REALIZADOS</i>	75
5.2.1 – <i>ROTINA PARA TESTES DO IMPORTADOR DE DADOS</i>	75
5.2.2 – <i>ROTINA PARA TESTES DA CENTRAL DE MONITORAMENTO SEM PERCURSO</i> <i>DEFINIDO</i>	79
5.2.3 – <i>ROTINA PARA TESTES DA CENTRAL DE MONITORAMENTO COM PERCURSO</i> <i>DEFINIDO</i>	81
5.3 – <i>RESULTADOS DA APLICAÇÃO DO MODELO</i>	82
CAPÍTULO 6 - CONCLUSÃO	83
6.1 – <i>CONCLUSÕES</i>	83
6.2 – <i>SUGESTÕES PARA TRABALHOS FUTUROS</i>	84
REFERÊNCIAS BIBLIOGRÁFICAS	85
APÊNDICE A – CÓDIGO DO IMPORTADOR DE DADOS	88
APÊNDICE B – DISPONIBILIZAÇÃO DO CÓDIGO FONTE DO PROJETO	95
APÊNDICE C – CÓDIGO PARA CRIAÇÃO DA BASE DE DADOS	96
ANEXO I – MANUAL DE INSTALAÇÃO DO FRAMECALIXTO	121

LISTA DE FIGURAS

Figura 1 – Visão Geral do Projeto.....	17
Figura 2 - A constelação GPS – Fonte: http://www.defenseindustrydaily.com/the-gps-constellation-now-and-future-01069/	26
Figura 3 - Funcionamento do GPS – Fonte: http://www.nwm.com.br/tarik/amp/gps.htm	26
Figura 4 - Segmentos GPS – Fonte: adaptado de: http://www.overcar.com.br/como-funciona-o-gps/	28
Figura 5 - Recepção do Sinal GPS – Adaptado de Thienne M. Johnson, 2000.....	29
Figura 6 - Método de Triangulação – Adaptador de Thienne M. Johnson, 2000.....	29
Figura 7 – Exemplo de erros inerentes ao sistema GPS – Fonte: adaptado de http://www.cerne-tec.com.br/TutorialGPS.pdf	31
Figura 8 – O globo terrestre e suas linhas de latitude e longitude – Fonte: adaptado de: http://www.jrank.org/history/pages/8345/How-Do-I-Use-Globes-Maps.html	32
Figura 9 - GPS Tracker TK-102 – Fonte: Xexun, 2010	35
Figura 10 – Estatística de uso do PHP – Fonte: Netcraft, 2007.	40
Figura 11 – Evolução do GSM – Fonte: Adaptado de [REDES GSM, GPRS, EDGE E UMTS, 2010].....	41
Figura 12 – Estatísticas de uso de softwares servidores WEB – Fonte: Netcraft, 2010.....	44
Figura 13 – Módulo TK-102 com o CHIP GSM instalado.	52
Figura 14 – Primeira mensagem de configuração do módulo GSM/GPRS/GPS.....	53
Figura 15 - Segunda mensagem de configuração do módulo GSM/GPRS/GPS	54
Figura 16 - Terceira mensagem de configuração do módulo GSM/GPRS/GPS.....	54
Figura 17 – Configuração para a portadora de dados VIVO no módulo GSM/GPRS/GPS. ...	56
Figura 18 – Exemplo de mensagem enviada pelo módulo TK-102	57
Figura 19 – Array mapeado com as informações provenientes do módulo TK-102.....	58
Figura 20 – Estrutura de pastas do sistema.....	59
Figura 21 - Modelo Entidade Relacionamento do projeto	61
Figura 22 – Caso de uso principal do sistema, atores: Administrador e Monitorador.	62
Figura 23 – Menu de acesso aos cadastros, para o usuário com perfil de monitorador.	63
Figura 24 – Menu de acesso aos cadastros, para o usuário com perfil de monitorador.	63
Figura 25 – Formulário de Cadastro de Pessoas.....	64
Figura 26 – Formulário de Cadastro de Veículos.....	65
Figura 27 - Formulário de Cadastro dos Detalhes do Veículo	65

Figura 28 – Formulário de Cadastro dos Tipos de Local.	66
Figura 29 – Formulário para o Cadastro de Locais	67
Figura 30 – Formulário para Cadastro de Percursos.	67
Figura 31 – Formulário de Cadastro para o Tipo de Ocorrência.....	68
Figura 32 – Formulário de Cadastro para Ocorrências.....	69
Figura 33 – Formulário de Cadastro de Encomendas no Sistema.	70
Figura 34 – Diagrama de atividades para o Cadastro de Encomendas.....	70
Figura 35 – Central de Monitoramento do Projeto.....	71
Figura 36 – Fórmula de haversine	73
Figura 37 – Terminal executando o importador de dados do sistema.....	76
Figura 38 – Dados inseridos no banco de dados a partir de mensagens recebidas do módulo TK-102.....	77
Figura 39 – Informações inseridas no banco, obtidas por intermédio do módulo TK-102.....	78
Figura 40 – Veículo sendo rastreado por intermédio da central de monitoramento.....	79
Figura 41 – Veículo monitorado por intermédio da Central de Monitoramento.....	80
Figura 42 – Modelo de rastreamento por um percurso previamente cadastrado.....	81

LISTA DE TABELAS

Tabela 1 - Comparação dos sistemas GNSS	22
Tabela 2 - Vantagens e desvantagens do uso de GPS em sistemas de Rastreamento de Veículos.....	27
Tabela 3 - Erros inerentes ao posicionamento GPS	30
Tabela 4 - Significado dos campos da mensagem \$GPRMC.....	34
Tabela 5 - Especificações do GPS Tracker TK-102.....	36
Tabela 6 - Estimativa de Custos do Projeto.....	51

LISTA DE SIGLAS E ABREVIATURAS

ABREVIATURAS

Etc. por et cetera

Obs. por observação

SIGLAS

WEB ou **WWW** *World Wide Web.*

GPS *Global Positioning System.*

GSM *Global System for Mobile Communications.*

GPRS *General Packet Radio Service.*

PHP *PHP Hypertext PreProcessor.*

URL *Uniform Resource Locator.*

HTTP *Hypertext Transfer Protocol.*

JS *JavaScript*

CSS *Cascading Style Sheets.*

KML *Keyhole Markup Language.*

POO *Programação Orientada a Objetos.*

IP *Internet Protocol*

SGBD *Sistema de Gerenciamento de Banco de Dados*

SQL *Structured Query Language*

RV *Rastreamento de Veículos*

NMEA *National Marine Electronics Association*

ASCII *American Standard Code for Information Interchange*

GPGSA *GPS DOP and active satellites*

GPGSV *GPS Satellites in View*

GPGLL *Geographic position, Latitude / Longitude*

GPGBA *Global Positioning System Fix Data*

GPRMC *Recommended Minimum specific GPS/Transit data*

HTML *HyperText Markup Language*

XHTML *eXtensible Hypertext Markup Language*

DHTML *Dynamic HTML*

PC *Personal Computer*

DOM *Document Object Model*

GPL *General Public License*
SMS *Short Message Service*
MVC *Model View Controller*
API *Application Programming Interface*
TSP *Traveling Salesman Problem*
SSH *Secure Shell*
IMEI *International Mobile Equipment Identity*

RESUMO

A crescente integração de tecnologias aplicadas ao rastreamento de veículos está em clara expansão de funcionalidades. Esta integração de tecnologias possibilita o acompanhamento sistemático do posicionamento do veículo. A determinação deste posicionamento engloba alguns variados tipos de exatidão e a sua obtenção em intervalos de tempo previamente definidos.

Este trabalho tem como objetivo a implementação de um sistema que forneça informações sobre o posicionamento de um veículo em um dado momento. O protótipo de sistema de rastreamento desenvolvido poderá ser adotado em empresas do ramo de transporte. Por intermédio de um módulo GPS integrado a um modem GSM as informações referentes ao posicionamento, velocidade e altitude do veículo obtidas são enviadas ao servidor através do serviço GPRS em um intervalo de tempo pré-definido. O servidor possui um IP fixo e uma porta liberada para que esta comunicação seja estabelecida. Um script PHP denominado "Importador de Dados" é o responsável pelo tratamento das informações e a alimentação da tabela inerente ao rastreamento, no banco de dados.

Do lado do servidor, está instalado o sistema nomeado de iGeoRastreador, solução baseada em mapas sob demanda para viabilizar a tarefa de rastreamento de veículos. Há duas opções para que seja possível o rastreamento: Cumprimento de percursos previamente cadastrados para entrega de encomendas ou rastreamento livre, onde o veículo será acompanhado na central de monitoramento por um funcionário da empresa, apenas com o intuito de se ter controle sobre o deslocamento do mesmo.

A solução proposta é implementada utilizando os recursos da linguagem PHP 5 e do SGBD Postgresql, para armazenamento das informações do sistema além de outras bibliotecas auxiliares.

Palavras Chave: módulo GPS/GSM/GPRS, Rastreamento de veículos, Tecnologia GSM, GoogleMaps, Navegação por satélite, Coordenadas Geográficas

ABSTRACT

The growing integration of technologies applied to vehicle tracking is in clear expansion of features. This integration technology enables the systematic monitoring of the positioning of the vehicle. The determination of this position encompasses several different types of accuracy and collection from pre-defined time intervals.

This paper aims to implement a system that provides information about the positioning of the vehicle in a given time. The prototype tracking system developed can be adopted in companies in the field of transport. Through an integrated GPS module to a GSM modem information regarding the position, speed and altitude of the vehicle obtained are sent to the server using the GPRS service in a range of pre-set time. The server has a fixed IP and a port released for such communication to be established. A PHP script called "Data Importer" is the controller of the information and the power inherent in the tracking table in the database.

On the server side, the system is installed named iGeoRastreador, solution-based maps on demand to facilitate the task of tracking vehicles. There are two possible options for screening: Meeting paths for previously registered parcel tracking or free, where the vehicle will be accompanied at monitoring station by an officer of the company, just in order to have control over the displacement of same.

The proposed solution was implemented using the features of PHP 5 and PostgreSQL DBMS for storage of information system and other auxiliary libraries.

Key Words: module GPS / GSM / GPRS Vehicle tracking, GSM technology, Google Maps, Satellite Navigation, Geographic Coordinates

CAPÍTULO 1 – INTRODUÇÃO

1.1 – APRESENTAÇÃO DO PROBLEMA

Empresas de transporte têm seus veículos como seu principal ativo. Inserida em um contexto de competitividade, a empresa se faz refém de estar constantemente inovando para que a mesma sobressaia dentre tantas outras. Para tanto, se faz necessário que a empresa possua uma estrutura e uma logística que possibilitem obter informações operacionais no menor tempo possível e assim traçar novas estratégias para a competição no mercado.

Neste mesmo enfoque, podemos destacar também o roubo de cargas e veículos em constante aumento no Brasil. Sistemas de rastreamento via satélite podem ser empregados para resguardar o patrimônio da empresa desta modalidade de crime. A utilização deste sistema tem se mostrado eficiente para inibir ações de quadrilhas organizadas. A adoção deste tipo de sistema está em ascensão por parte das empresas do ramo e com isso, novos sistemas com características mais robustas e que englobem melhores funcionalidades devem ser desenvolvidos para esta demanda do ramo de transportes.

Sendo assim, este projeto propõe o desenvolvimento de uma nova ferramenta que auxilie as empresas do ramo de transporte a gerenciar sua frota, com o intuito de rastrear sua frota e com a funcionalidade de fornecimento de informações para que sejam traçadas melhores estratégias operacionais e logísticas, visando também a redução de furtos de cargas.

Seria possível a aplicação de uma ferramenta para monitoramento por intermédio de mapas que forneça informações para melhoria logística da empresa e também auxiliar em eventos inesperados como o furto de mercadorias?

1.2 – OBJETIVOS DO TRABALHO

O objetivo geral do trabalho é implementar e integrar a solução de hardware e software que permita monitorar por intermédio de mapas sob demanda, uma frota de veículos providos de um dispositivo transmissor GPS, sendo assim possível a identificação de sua localização e outras informações operacionais. Através destas informações poderão ser traçadas melhores estratégias operacionais e monitoramento quanto a ocorrência de furto de mercadorias e outras.

Como objetivos específicos podemos enumerar:

- Integração do módulo GPS/GSM/GPRS (GPS Tracker TK-102) ao sistema iGeoRastreador desenvolvido neste projeto por intermédio de um “Importador de Dados” que fará a inserção dos dados provenientes do GPS no banco;
- Integração dos módulos de gerenciamento e central de monitoramento para manutenção das informações de rastreamento do veículo e também informações diversas como velocidade e avisos de pane ou outras ocorrências.
- Desenvolvimento do módulo administrativo (cadastros auxiliares) integrado ao módulo de monitoramento;
- Desenvolvimento do módulo dos mapas sob demanda (central de monitoramento);

1.3 – JUSTIFICATIVA E IMPORTÂNCIA DO TRABALHO

Este trabalho possui importância para a o rastreamento online dos veículos de empresas do ramo de transporte. O projeto propõe um protótipo de um sistema que auxilia no monitoramento dos veículos em empresa de transporte e também no fornecimento de dados para sua logística operacional. Podemos citar também o auxílio na área de segurança do transporte das mercadorias bem como dos motoristas que efetuam o transporte, visto que um funcionário poderá acompanhar em tempo integral, por intermédio da central de monitoramento o veículo, caso ocorra algum evento não previsto durante o percurso da entrega o mesmo poderá tomar as ações necessárias. A partir das informações de posicionamento do veículo fornecidas pelo módulo GSM/GPRS/GPS escolhido para o projeto, será possível rastrear o veículo e acompanhar online todos os percursos que o motorista da empresa está praticando com o veículo rastreado.

1.4 – ESCOPO DO TRABALHO

Neste projeto, é realizada a integração de um aparelho GPS que possui um módulo GSM/GPRS embutido a um sistema de rastreamento de veículos que utiliza mapas sob demanda para realização desta tarefa. O aparelho envia as informações do veículo (coordenadas geográficas, velocidade, etc) a um servidor WEB onde há um script “Importador de Dados” que recebe as informações e efetua o tratamento dos dados. Os dados são armazenados no banco de dados do projeto, onde o sistema principal *iGeoRastreador* pode fazer uso destas informações. O aparelho GPS funciona enviando os dados da localização do veículo de tempos em tempos para o servidor.

Por intermédio do sistema *iGeoRastreador*, que possui uma central de monitoramento baseada em mapas sob demanda, o operador do sistema terá acesso a informações de tempos em tempos em um intervalo de tempo pré-determinado, dos veículos da empresa. Na figura 1 é apresentada uma visão geral da integração proposta.

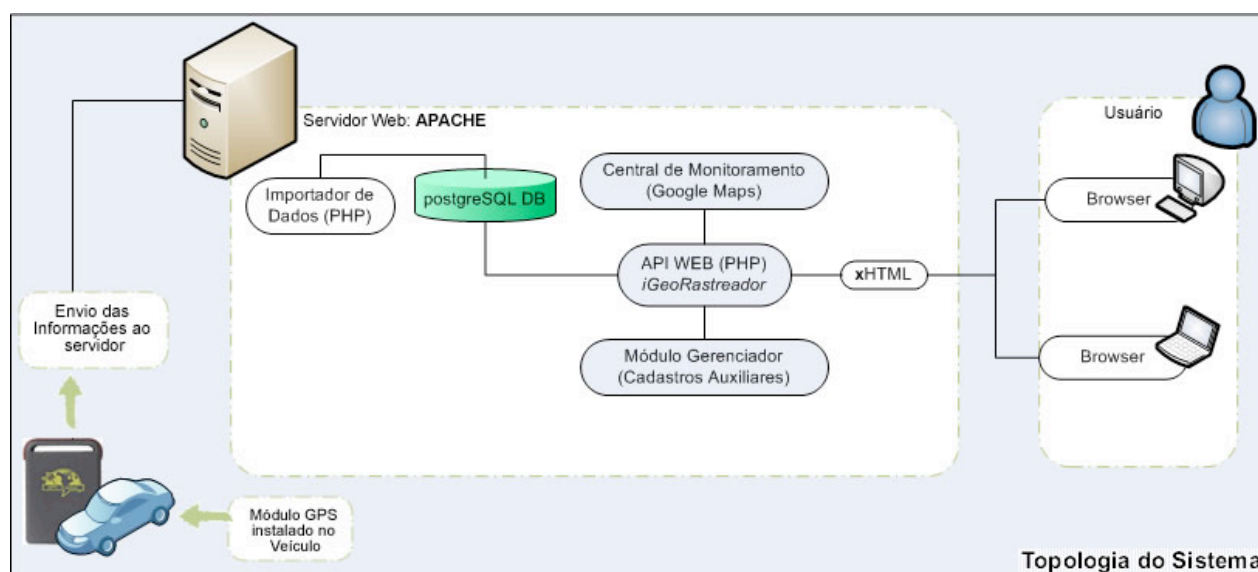


Figura 1 – Visão Geral do Projeto.

No sistema há a necessidade de um servidor WEB com o endereço de IP fixo e uma porta específica liberada no firewall, para que o GPS possa por intermédio da rede GSM sobre o serviço GPRS enviar as informações necessárias para o funcionamento do sistema *iGeoRastreador*. O GPS envia as informações e o script importador de dados efetua o tratamento da informação recebida e posteriormente alimenta a base de dados do sistema. Este é um sistema que poderá ser acessado por intermédio de plataformas como o *browser* de um notebook ou até mesmo de um celular que possua um web browser compatível.

1.5 – RESULTADOS ESPERADOS

Efetuar a integração do módulo GPS ao sistema, com o envio dos dados para um servidor WEB e desenvolver um software com a característica de central de monitoramento baseada em mapas sob demanda, que utilizem as informações obtidas através do módulo GPS.

O sistema deve prover um ambiente onde os operadores consigam acompanhar em tempo real o dia-a-dia operacional de uma empresa de transporte. O software desenvolvido pode se tornar uma ferramenta que proporcione um diferencial para a melhor organização operacional e logística da empresa.

1.6 – ESTRUTURA DA MONOGRAFIA

Este projeto está composto de 6 capítulos que visam expor de maneira clara, tópicos inerentes ao desenvolvimento da solução proposta. Segue abaixo a descrição dos aspectos abordados na pesquisa e implementação no decorrer deste projeto.

No capítulo 1 temos a introdução do projeto. Neste capítulo é explicado o problema encontrado e o que está sendo proposto para se solucionar o mesmo.

No capítulo 2 é apresentado ao leitor detalhadamente o problema. Por intermédio de pesquisas são efetuadas comparações com soluções já existentes e um cenário atual é comentado neste capítulo.

No capítulo 3 as tecnologias aplicadas no projeto são expostas para que o leitor se familiarize com as mesmas e para um melhor entendimento da implementação, posteriormente.

No capítulo 4 temos uma explicação generalizada do que será desenvolvido.

No capítulo 5 temos todo o desenvolvimento do protótipo, tanto a parte de hardware quanto a de software. A modelagem realizada, os requisitos da aplicação, bem como os recursos necessários para seu funcionamento e os testes realizados são descritos neste capítulo. Também são expostas algumas telas da aplicação e descrição detalhada de como foram construídas as mesmas.

No capítulo 6 é apresentada a finalização da pesquisa com as conclusões obtidas no decorrer do desenvolvimento do projeto. Também são citadas melhorias que podem ser aplicadas ao projeto no futuro e não foram implementadas neste momento pois fugiriam do escopo inicial do projeto.

CAPÍTULO 2 - APRESENTAÇÃO DO PROBLEMA

O real crescimento da demanda por sistemas de rastreamento de veículos está aquecendo a concorrência entre as empresas do ramo de transporte. A tendência, é que as empresas, cada vez mais se preocupem em manter e oferecer melhores serviços, visando qualidade e segurança da mercadoria transportada, visto que a entrega do produto na hora marcada e com o menor custo é vital para a competitividade [Segalla, 2006].

Para isto, se faz necessário que estas empresas utilizem um sistema que auxilie na tomada de decisões (gerenciamento de riscos), logística e operação da frota. Neste caso, o sistema de rastreamento integrado a um ERP pode ser aplicado.

O rastreamento de veículos visa ao fornecimento e ao gerenciamento sistemático da posição e do estado dos veículos, com variados níveis de exatidão e intervalos de tempo. É empregado para aumentar a eficiência dos despachos, otimizar o uso dos veículos, reduzir o custo operacional e atender às exigências das seguradoras, uma vez que aumenta a segurança das cargas, do casco e dos motoristas [Rodrigues et al, 2009].

As planilhas, formulários e documentos operacionais utilizados atualmente para o controle das informações operacionais da empresa muitas vezes não dispõe da informação correta, o que acarreta num resultado final incompleto, com relatórios apresentando informações incoerentes ou erradas.

Aparelhos que acumulam a informação (rastreamento passivo) para que seja descarregada após o término da operação juntamente ao terminal, não contam com informações online para os gestores ou funcionários operadores da empresa, informações estas que poderiam ser utilizadas para garantir a segurança no ato do transporte da mercadoria.

Já o sistema que alimenta o banco de dados após a operação, não conseguirá informar online os resultados dos eventos que foram realizados ou que estão em andamento.

Com um sistema de rastreamento online por intermédio de mapas sob demanda, as planilhas poderão ser suprimidas e a quantidade de gastos com mão de obra e planilhas de papel poderão ser reduzidos, visto que não serão mais necessários os digitadores para alimentar o banco de dados do sistema, apenas os operadores irão controlar as operações.

Um sistema de rastreamento de veículos, visa combinar a instalação de um módulo receptor GPS eletrônico dotado de modem GSM transmissor de informações em um veículo ou frota de veículos, com um software em formato de central de monitoramento para permitir que funcionários da empresa exerçam controle da rotina operacional dos veículos, tendo em

vista que a informação da localização do veículo será disponibilizada por intermédio da navegação por satélite.

Outros módulos podem ser agregados ao sistema adicionando funcionalidades importantes, como o bloqueio do veículo, botão de aviso de pânico, sensor para avaliação de consumo, controle de velocidade e inúmeros outros que podem ser adotados dependendo do ramo em que a empresa atua, porém estas agregações de componentes ou sensores não fazem parte do escopo do projeto proposto.

Após a obtenção da informação sobre a localização do veículo, será possível por intermédio da central de monitoramento acompanhar os locais visitados por ele, em mapas sob demanda através da Internet.

O sistema permite ao proprietário da empresa através da Internet, planejar, gerenciar, fiscalizar e aferir os deslocamentos efetuados por cada um dos seus veículos, durante sua operação no dia.

Segundo [Rodrigues et al, 2009], em um cenário de contínuo desenvolvimento tecnológico, os custos da implementação tornam-se a cada dia mais baixos e simultaneamente, ampliam-se as funcionalidades e opções de monitoramento de veículos. Os aspectos mais importantes para a adoção de um sistema de RV são:

- Sistema de comunicação, equipamentos embarcados e sistema de gestão.
- Especificação dependente do tipo de operação do gestor da frota, do modelo de monitoramento espacial, do tipo de carga e das características do deslocamento.
- Os atributos mais importantes do monitoramento são a exatidão das coordenadas geográficas, frequência da obtenção e transmissão de dados, além do custo associado.

Embora existam várias soluções integradas para monitoramento de frotas, destaca-se o grande número de sistemas do gênero que possuem suas funcionalidades disponibilizadas via web [Rodrigues et al, 2009].

CAPÍTULO 3 – REFERENCIAL TEÓRICO

3.1 – RASTREAMENTO DE VEÍCULOS (RV)

O RV é um sistema de monitoramento que gerencia a localização e o estado de um veículo a cada momento, enquanto ele se desloca sobre a superfície terrestre [Rodrigues et al, 2009].

Os sistemas de RV, tiveram suas operações iniciadas ao final da década de 1960. Denominados inicialmente de Automatic Vehicle Location (AVL), a sua principal funcionalidade era relacionada ao auxílio à gestão de tráfego. A utilização do sistemas de RV também visa o gerenciamento de logística e de risco. Os primeiros sistemas RV foram implantados no Brasil com o intuito de prevenir roubos de mercadorias durante o transporte e também a recuperação destes bens.

Atualmente, as funcionalidades destes sistemas tem sido ampliadas para que através dos mesmos, informações relevantes, além do posicionamento do veículo com a mercadoria, possam ser obtidas. As informações sobre a posição do veículo e outros dados que podem ser coletados permitem a realização de operações associadas, de acordo com a demanda de cada usuário e o tipo de operação (controle logístico, controle de risco, gerenciamento de frotas, gestão de transporte público e outros) [Rodrigues et al, 2009].

3.2 – SISTEMA GLOBAL DE NAVEGAÇÃO POR SATÉLITE - GNSS

Sistema Global de Navegação por Satélite (*GNSS*, acrônimo em inglês para Global navigation satellite system) é o nome genérico utilizado para denominar os sistemas de navegação por satélite que permitem a determinação de posicionamento geoespacial com cobertura global.

Por intermédio de uma constelação de satélites dotados de receptores eletrônicos, emissores de sinais, se torna possível identificar com certa precisão sua localização em termos de latitude, longitude e altitude. A constelação de satélites, é a responsável por enviar sinais de rádio que possibilitam a realização desta orientação. Há algum tempo atrás, apenas o *GPS* figurava no cenário dos *GNSS*. Outros países possuem atualmente, iniciativas semelhantes no âmbito de sistemas de navegação por satélite. A Rússia possui o sistema *Glonass* (Já em funcionamento), a Europa lançou o *Galileo* e a China possui planejamento para o lançamento

dos satélites do sistema *Beidou*, que não será citado neste trabalho devido a falta de informações concretas sobre seu desenvolvimento.

Tabela 1 - Comparação dos sistemas GNSS

	GPS	GLONASS	GALILEO
Número de Satélites	21 + 3	21 + 3	27 + 3
Número de Planos Orbitais	6	3	3
Altitude	20.200km	19.100km	23.616km
Período de rotação	11h:58min	11h:15min	14h:07min
Inclinação ao plano	~55°	~64°	~56°
Peso do Satélite	1.110kg	1.400kg	700Kg
Área do Painel Solar	14m ²	23m ²	13m ²

Fonte – Adaptado de <http://www.positim.com>.

Os satélites artificiais, componentes do sistema de navegação por satélite, possuem em sua estrutura relógios atômicos utilizados para executar com precisão e exatidão a tarefa de obter informações requeridas a estes sistemas. Relógios atômicos são capazes de fornecer informações na ordem dos nano segundos (10^{-9} s). Para tanto, todos os satélites da constelação devem necessariamente estar com os relógios atômicos sincronizados entre si.

Os *GNSS* foram concebidos inicialmente para fins militares, porem após vários anos ganharam utilidade em outras áreas específicas. Ainda possuem características para o uso militar porem são aplicados na navegação aérea, rastreamento de veículos, sistemas de localização para emergência, agricultura e etc.

Para o projeto proposto, utilizou-se o *GPS* como o sistema que auxilia na obtenção das informações referentes aos veículos da empresa de transporte, visto que é considerado o sistema mais completo atualmente e amplamente difundido.

3.2.1 – O SISTEMA GLONASS

O sistema GLONASS é o sistema alternativo de navegação por satélite, desenvolvido pelos Rússia. Denominado Global'naya Navigatsionnaya Sputnikovaya Sistema (Sistema de Satélites de Navegação Global), este sistema possui várias características do seu funcionamento semelhantes às do sistema americano. Trata-se de um sistema composto de 24 satélites (21 ativos e 3 de reposição) operando na órbita de 19.100km a uma inclinação de 64,8 graus, e seu período de rotação de aproximadamente 11 horas, também transmitindo informações sobre relógio e efemérides [Rodrigues et al, 2009].

Composto por três níveis orbitais, cada um desses três níveis conta com oito satélites. Este sistema foi projetado para possuir a mesma quantidade de satélites do sistema americano. Possui serviços de navegação diferenciados, tanto para o uso civil quanto para o uso militar, permitindo uma precisão na ordem de 60m na localização e para o relógio da ordem de 1 μ s. Atualmente opera apenas parcialmente devido à falta de investimento do governo ante sua situação econômica e também a falhas ocorridas nos satélites lançados. Por isso, ainda é um sistema que possui certos índices de falhas na precisão das informações e possui seu uso praticamente restrito ao governo russo.

Segundo Rodrigues, Cugnasca e Filho (2009) as informações sobre posicionamento de uso civil e militar são transmitidas pelos satélites em frequências separadas, na região do espectro de micro ondas. A informação civil usa a frequência L1, de 1.602 MHz. Diferentemente do sistema GPS, o Glonass usa divisão de frequência em canais individuais de banda fixa de 562,5 kHz para cada satélite, começando na frequência de 1.602 MHz, para modulação do sinal com as informações de navegação.

A utilização do sistema GLONASS juntamente a outros sistemas GNSS permitirá elevar consideravelmente o nível dos sistemas de navegação por satélite relativos a precisão, acesso, integridade e continuidade dos serviços de navegação para as diversas áreas que a utilizam.

Apesar de fontes de informação não darem como certas a continuidade do projeto, o governo russo firmou parecerias internacionais que estão permitindo que isso aconteça. Estes acordos garantem o auxílio financeiro para que o projeto siga em desenvolvimento. Novos modelos de satélites vêm sendo desenvolvidos, com características aprimoradas, com menor peso, maior vida útil e maior capacidade de transmissão. A conclusão do sistema está prevista para 2012. As especificações do sistema Glonass servirão como base para o novo sistema de navegação europeu, atualmente em desenvolvimento [Rodrigues et al, 2009].

3.2.2 – O SISTEMA GALILEO

O projeto europeu para sistemas de navegação global por satélite é denominado Galileo. Este é o único projeto de GNSS concebido para fins civis e não militares. O mesmo será inter-operável aos outros dois sistemas GNSS existentes: GPS e o GLONASS. Ele oferece vantagens, reunindo características do sistema GPS e também o Glonass, possuindo uma constelação de 30 satélites sendo 3 deles sobressalentes, efetuando assim maior cobertura do globo nas regiões polares. No Galileo, a disponibilidade do serviço é garantida e nas circunstâncias mais extremas ele informará ao usuário em questão de segundos a ocorrência de falha em um dos satélites. Estas características tornarão o Galileo adequado a aplicações onde segurança é crucial.

O primeiro satélite experimental do GNSS Galileo, GIOVE-A lançado em 28 de dezembro de 2005, com o objetivo de consolidar tecnologias críticas já desenvolvidas. O segundo satélite (GIOVE-B) foi lançado em 27 de abril de 2008. O objetivo destes dois satélites é consolidar tecnologias críticas ainda em desenvolvimento. O GIOVE-B carrega em seu interior, o maser de hidrogênio passivo, um relógio ultra-estável que pode ser a grande promessa no âmbito de sistemas GNSS futuros [Galileo Information Center, 2010].

Após a conclusão da implantação do Galileo, o mesmo estará composto por 30 satélites sendo 27 operacionais e 3 de reposição. Os satélites serão lançados em órbitas de 23.616km (acima dos satélites do GPS e Glonass), dispostos ao longo de três planos orbitais. Para o futuro, existe a previsão de que serão posicionados mais 8 satélites geoestacionários nas proximidades do equador, para o ganho de precisão de posicionamento nesta área. Ao final, serão totalizados 38 satélites [Rodrigues et al, 2009].

Com a mesma variedade de informações moduladas em três portadoras de frequências L1, L2 e L3 (1.575,42 MHz, 1.227,60 MHz e 1.381,05 MHz respectivamente) do sistema americano, o Galileo envia informações de uso civil nas duas primeiras frequências, de forma redundante, as quais podem ser combinadas para o refinamento da posição calculada. As frequências L4 e L5 (1.841,40 MHz e 1.176,45 MHz) também foram incorporadas ao projeto, e a frequência L5 é utilizada para o sistema de sinalização de emergência [Rodrigues et al, 2009].

O período de rotação dos satélites é de aproximadamente 14 horas. A inclinação dos planos orbitais será de 75 graus. O grande número de satélites em conjunto com a otimização da constelação, e da disponibilidade de três satélites ativos de reposição, irá garantir que a perda de um satélite não afete os usuários finais do sistema.

Operacional desde 2008, o GNSS Galileo auxiliará na resolução de problemas referentes a mobilidade e transporte que convivem atualmente as varias regiões do planeta. A data prevista para o início da operação completa do sistema Galileo é o ano de 2013 [Rodrigues et al, 2009].

3.2.3 – SISTEMA DE POSICIONAMENTO GLOBAL - GPS

Com o seu projeto iniciado em 1973, o sistema denominado Navigation Satellite Timing and Ranging Global Positioning System (Navstar GPS) mais conhecido como GPS, teve sua concepção por intermédio de fundos do Departamento de Defesa dos Estados Unidos para fornecer a posição instantânea e a velocidade de um ponto sobre a superfície terrestre, ou próximo a ela. Seu funcionamento é considerado pleno desde 1995. Atualmente, o GPS é utilizado por diversos segmentos da sociedade e seu uso está a cada dia mais difundido.

Até hoje o GPS é a mais sofisticada, engenhosa e eficiente ferramenta que o Homem criou para saber sua posição na Terra. Desde 1973, estima-se que o Governo norte-americano tenha investido algumas dezenas de bilhões de dólares no projeto. Nesse período, algumas alterações de ordem técnica foram introduzidas, e os constantes aprimoramentos da tecnologia tem provocado um progressivo aumento na precisão. Hoje, já é possível obter-se uma precisão da ordem de centímetros para a posição de um determinado ponto [Cugnasa, 2007].

O sistema GPS permite ao usuário a obtenção de seu posicionamento (latitude, longitude e altitude) em tempo real. O sistema é composto por 24 satélites sendo 3 deles de reserva, onde todos estão distribuídos em seis planos orbitais. A uma altura de 20.200 km, os satélites possuem uma inclinação de 55 graus em relação ao equador e a aproximadamente 12 horas completam uma volta na Terra. Os satélites possuem suas operações sincronizadas para que os sinais de repetição sejam transmitidos no mesmo instante de tempo. Realiza-se o cálculo das órbitas de maneira que independentemente do momento, ao menos quatro satélites se tornem visíveis em qualquer ponto do globo terrestre.

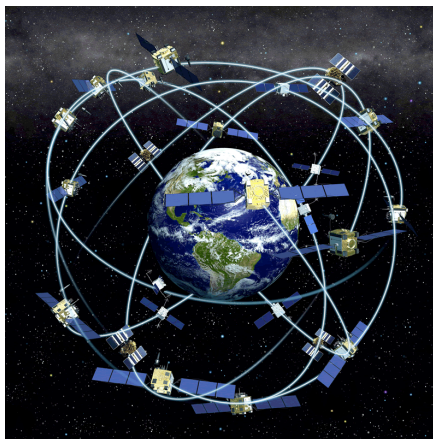


Figura 2 - A constelação GPS – Fonte: <http://www.defenseindustrydaily.com/the-gps-constellation-now-and-future-01069/>

Todos os satélites constituintes do sistema GPS são dotados de quatro relógios atômicos, dois de rubídio e dois de césio, onde o sincronismo entre eles é mantido com precisão de um microssegundo. Na frequência-base são utilizados os relógios de rubídio. Esta frequência é empregada no controle de todas as temporizações e geração das frequências de rádio para os sistemas internos do satélite. Apesar de sua concepção para uso militar, o GPS também envia informações para o uso civil porem em uma outra frequência de rádio.

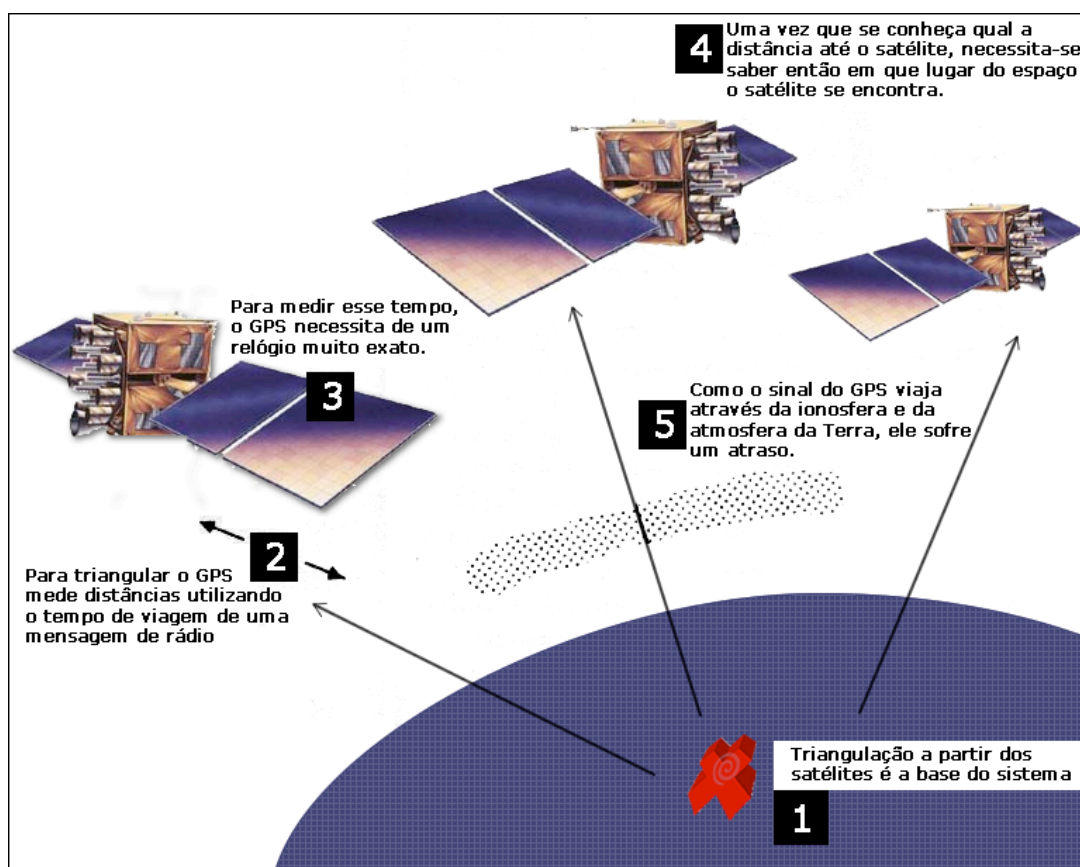


Figura 3 - Funcionamento do GPS – Fonte: <http://www.nwm.com.br/tarik/amp/gps.htm>

Os satélites transmitem continuamente sua localização e uma referência de tempo, por meio de ondas eletromagnéticas. A frequência do sinal está entre 1.227 MHz e 1.575 MHz. Como o sistema GPS é um sistema exclusivamente para envio de informações, onde os usuários apenas consomem informações enviadas pelos satélites o sistema é denominado passivo. Definiu-se assim o funcionamento do sistema, por questões de segurança. Os sinais captados pelos aparelhos receptores, são processados e transformados em coordenadas geográficas ou métricas.

Em resumo, o GPS determina o intervalo entre a transmissão e o recebimento do sinal para calcular a distância entre a transmissão e o recebimento do sinal para calcular a distância entre o receptor e o satélite. Após receber informações de pelo menos quatro satélites, a posição pode ser obtida por triangulação [Rodrigues et al, 2009].

Abaixo, uma tabela que dita vantagens e desvantagens mais significativas de se utilizar a tecnologia GPS para um sistema equivalente ao proposto neste projeto:

Tabela 2 - Vantagens e desvantagens do uso de GPS em sistemas de Rastreamento de Veículos

Vantagens	Desvantagens
Abrangência mundial	Não funciona em ambientes fechados (túneis, garagens e etc).
Uso consagrado (adotado em 80% dos veículos que utilizam rastreamento no Brasil).	Os sinais dos satélites podem ser obstruídos por pontes, viadutos, edifícios e matas.
Precisão de aproximadamente 5 metros (dependendo do receptor).	A geometria desfavorável dos satélites pode diminuir a precisão do posicionamento.
Não há custo de implantação e operação de rede.	O multicaminhamento, ou seja, a reflexão do sinal em algum objeto – como um edifício algo – interfere na precisão das coordenadas.
O preço dos equipamentos receptores diminui continuamente.	Os receptores, ao serem ligados ou após a perda de comunicação com o satélite, levam certo tempo para (re)iniciar a aquisição de dados (partida a frio).
A informação sobre a posição geográfica encontra-se no veículo.	

Fonte: adaptado de: Rastreamento de Veículos [Rodrigues et al, 2009].

3.2.3.1 – SEGMENTOS GPS

Por definição de projeto, o sistema GPS subdividiu-se em 2 segmentos: espacial e terrestre, que por sua vez pode ser subdividido em 2 subsegmentos: subsegmento de controle e subsegmento de usuário.

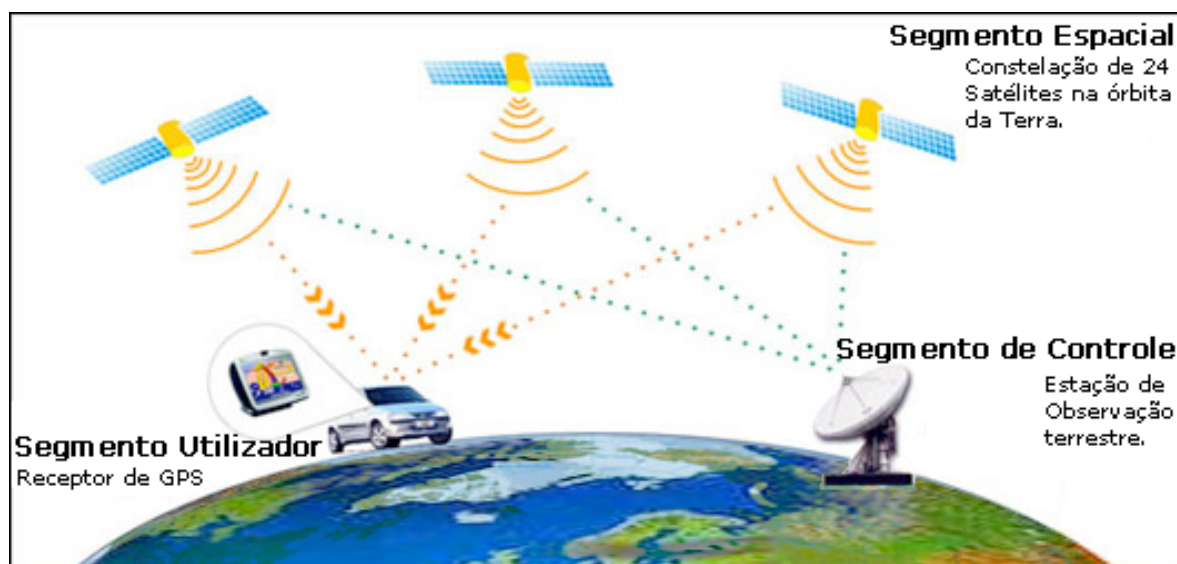


Figura 4 - Segmentos GPS – Fonte: adaptado de: <http://www.overcar.com.br/como-funciona-o-gps/>.

O segmento espacial é caracterizado pelos 24 satélites que estão em órbita nos seis planos orbitais a uma altitude de 20.000km. Estes satélites possuem 28 graus de visualização sobre a terra e estão inclinados a 55 graus em relação à linha do equador. Por alguns momentos, vários pontos da Terra são visualizados ao mesmo tempo por 6 a 10 satélites que focalizam a mesma área. Isto fornece redundância, pois apenas 4 satélites são necessários para uma determinação tridimensional de posição [Johnson, 2000].

A constelação dos 24 satélites GPS é controlada pelo segmento de Controle, na terra. O subsegmento de Controle do segmento terrestre é constituído por um conjunto de cinco estações espalhadas pelo globo que recebem continuamente a informação proveniente dos satélites e a repassam para a estação central, localizada no Colorado nos Estados Unidos. Toda a informação recebida é processada pelas estações onde os dados referentes a correção e sinais de controle são enviados aos satélites. Este é o segmento responsável pela monitoração do desempenho total do sistema onde por intermédio do mesmo a posição dos satélites pode ser corrigida e também pode haver a reprogramação do sistema com o padrão necessário.

O subsegmento de Usuário do segmento terrestre diz respeito aos receptores GPS e a comunidade de usuários. Os satélites transmitem em tempo real os sinais para os receptores

que por sua vez possuem uma unidade de processamento capaz de decodificar estes sinais em posição, velocidade e tempo estimado. O receptor, captando sinais de três satélites consegue calcular a posição em duas dimensões (longitude e latitude) de onde o mesmo se encontra. Com mais de três satélites, se torna possível o cálculo da posição em 3D (latitude, longitude e altitude). Como cada satélite envia sinais diferentes em intervalos de tempos diferentes, o receptor é capaz de efetuar os cálculos necessários para obtenção destas informações. São necessários no mínimo quatro satélites para que sejam calculadas as quatro dimensões: x , y , z (posição) e t (tempo) (Figura 4).

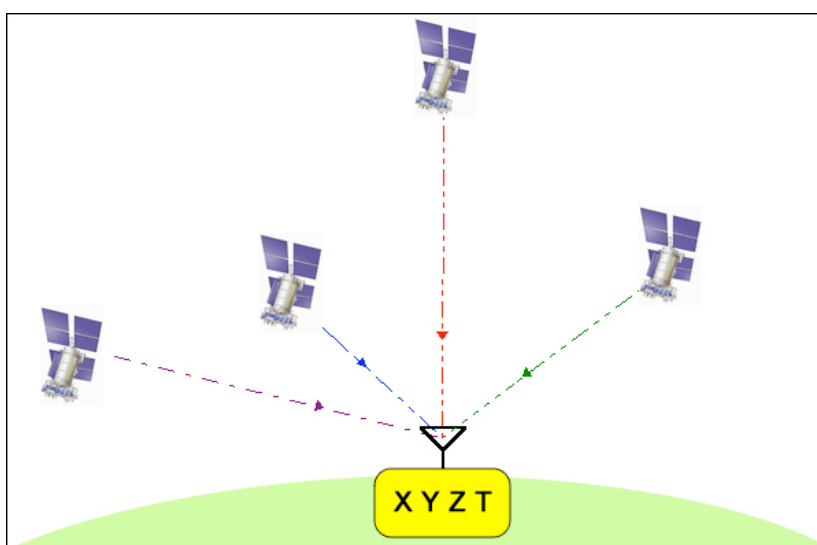


Figura 5 - Recepção do Sinal GPS – Adaptado de Thienne M. Johnson, 2000.

Por intermédio do método de triangulação, é possível determinar os valores de distância das três posições diferentes (x , y e z) (Figura 5).

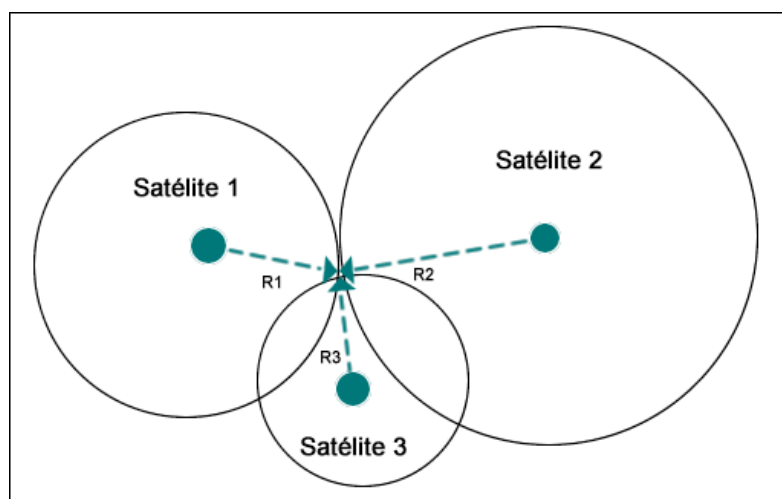


Figura 6 - Método de Triangulação – Adaptador de Thienne M. Johnson, 2000.

Conhecendo a distância em relação a três ou mais satélites, o receptor pode calcular a sua posição com base numa série de equações. Em teoria, a distância pode ser calculada multiplicando o tempo que o sinal demora a chegar pela velocidade a que este viaja (a velocidade da luz). No entanto, na prática são necessários cálculos mais sofisticados, uma vez que podem existir inúmeras interferências, como as condições atmosféricas ou as perturbações do Sol. A informação de três satélites é necessária para calcular a longitude e a latitude; no entanto, são necessários quatro para calcular também a altitude [Johnson, 2000].

Os receptores GPS utilizados atualmente possuem recursos os quais o tornam viáveis para utilização em diversas áreas, sendo utilizados também para pesquisas e afins.

3.2.3.2 – FONTES DE ERRO NO SISTEMA GPS

Mesmo sendo um sistema projetado para que se alcance a maior precisão possível na obtenção das informações pelos seus receptores, ainda existem fatores que afetam a precisão do GPS e ocasionam erros inerentes ao posicionamento.

Tabela 3 - Erros inerentes ao posicionamento GPS

Origem do Erro	Precisão
Ionosfera	5,5 metros
Efemérides	2,5 metros
Atraso no relógio	1,5 metros
Multicaminhamento	0,6 metros
Ruído no Receptor	0,3 metros
Erro Total	~5-15 metros

Fonte: Rodrigues et al, 2009.

A camada da terra conhecida como Ionosfera pode reduzir a velocidade do sinal do satélite no momento da travessia da atmosfera. Para corrigir este atraso, o GPS utiliza um mecanismo interno que efetua o calculo parcial do tempo comum de demora.

Erros de Efemérides são erros orbitais, que podem causar inexatidão do local informado do satélite.

Atrasos no relógio do receptor podem ocorrer, visto que os relógios portado por estes aparelhos não possuem a mesma precisão atômica do relógio instalado no satélite GPS. Estes erros são considerados erros de cronometragem muito leves.

O multicaminhamento é um erro produzido pela reflexão do sinal de GPS por algum objeto. Edifícios altos ou superfícies com grandes pedras podem refletir o sinal antes do receptor ser localizado. Isto interfere na precisão das coordenadas.

O receptor GPS com suas limitações e imperfeições também poderá ocasionar erro na obtenção das coordenadas. Ele está limitado à sua própria precisão, ou seja ao desvio padrão associado a cada medição.

Após a exposição das informações sobre os erros do GPS na Tabela 3, observa-se que mesmo com a existência de tais fatores que podem influenciar na precisão da obtenção das coordenadas, o sistema possui uma exatidão observada entre 5 e 15 metros.

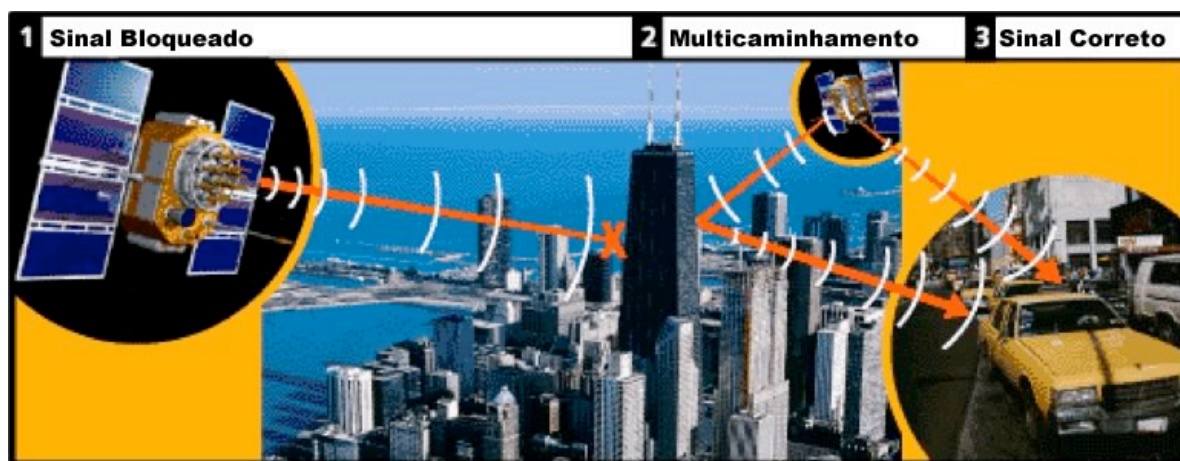


Figura 7 – Exemplo de erros inerentes ao sistema GPS – Fonte: adaptado de <http://www.cerne-tec.com.br/TutorialGPS.pdf>.

3.3 – COORDENADAS GEOGRÁFICAS

Segundo Winge (2010) o sistema referencial de localização terrestre é baseado em valores angulares expressos em graus, minutos e segundos de latitude (paralelos) e em graus, minutos e segundos de longitude (meridianos), sendo que os paralelos correspondem a linhas imaginárias E-W paralelas ao Equador e os meridianos a linhas imaginárias N-S, passando pelos polos, correspondentes a interseção da superfície terrestre com planos hipotéticos contendo o eixo de rotação terrestre.

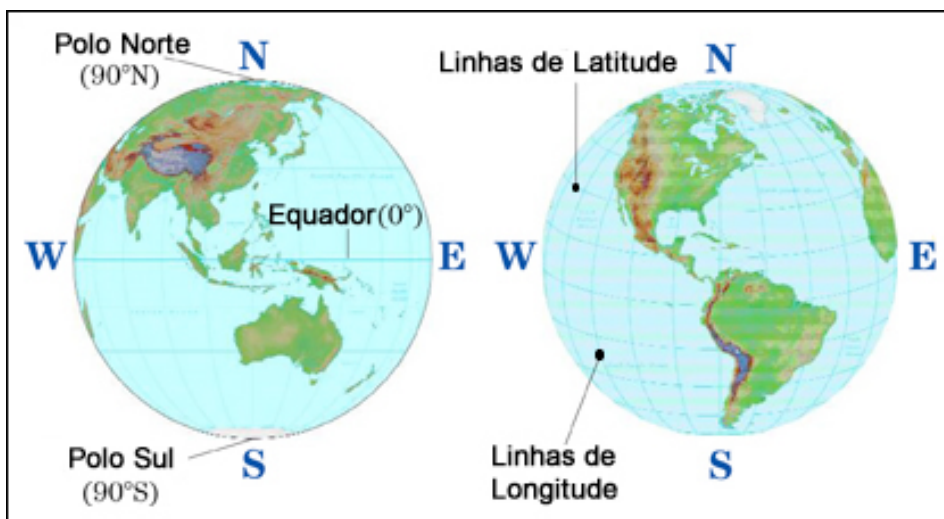


Figura 8 – O globo terrestre e suas linhas de latitude e longitude – Fonte: adaptado de:
<http://www.jrank.org/history/pages/8345/How-Do-I-Use-Globes-Maps.html>

Com as coordenadas geográficas se torna possível a definição de qualquer ponto no globo terrestre. São linhas imaginárias verticais (longitude) e horizontais (latitude) que dividem a terra em grandezas de graus, minutos e segundos.

A latitude é para o plano equatorial um ângulo de um ponto sobre a superfície da Terra, medido a partir do centro da esfera. Os paralelos são as linhas que unem pontos da mesma latitude. Essas linhas traçam círculos concêntricos na superfície da Terra, paralelos ao equador. O polo norte está a 90° N e o polo sul está a 90° S. O paralelo 0° de latitude é o chamado de equador. Considera-se o equador, o plano fundamental para todos os sistemas de coordenadas geográficas onde o mesmo divide o globo terrestre em hemisférios Norte e Sul.

A longitude é a distância ao meridiano de Greenwich medida ao longo do Equador. É o ângulo leste ou oeste de um meridiano de referência entre os dois polos geográficos para outro meridiano que passa por um ponto arbitrário. Ao contrário da latitude, onde não é definida a linha do equador como um marco inicial natural, na longitude não há uma posição inicial natural para ser demarcada. O Meridiano de Greenwich foi adotado como o primeiro meridiano de conhecimento mundial em 1884 na International Meridian Conference.

Os valores em grau da longitude são decompostos em 60 minutos e subsequentemente em 60 segundos. O formado correto de se expressar os valores de longitude é: graus° minutos' segundos". Para localidades situadas no oeste, é empregado o sinal de negativo (-) a frente do valor. Em alguns casos, autores empregam as letras "E" e "W" para indicações de Leste e Oeste. O casamento de valores de latitude e longitude obtém como resultado preciso um ponto em qualquer posição do globo terrestre.

3.4 – PADRÃO NMEA

O protocolo NMEA 0183 é instituído pela National Marine Electronics Association para que mensagens utilizadas em embarcações sejam padronizadas. O padrão NMEA 0183 define, dentre outros, características elétricas do sinal, protocolo da transmissão de dados, sincronismo e formatos específicos de sentenças de transmissão. As mensagens são constituídas por até 82 caracteres ASCII por padrão. Mensagens são transmitidas seguindo a seguinte especificação:

\$GP<identificador da mensagem><dados><*checksum><CR><LF>

Todas as mensagens devem ser precedidas do campo "\$GP" seguidas pelo identificador da mensagem que pode ser do tipo GSA, GSV, GLL, GGA e RMC dentre vários outros que surgem de acordo com necessidades específicas. Demais campos são separados por vírgulas. Ao final, a mensagem é acrescida de um caractere de retorno <CR> e de uma nova linha <LF>. Campos que não possuem informação, ou seja, nulos, também são separados por vírgulas porem omitidos.

O campo de checksum (índice de checagem de integridade da mensagem) indica por intermédio de um algoritmo a ocorrência de erro no envio da mensagem. O mesmo é precedido pelo caractere "*".

Grande maioria dos aparelhos receptores de GPS aceitam como padrão o protocolo NMEA, adotado por iniciativa de seus fabricantes por ser um padrão consolidado. Ele é um padrão de uso voluntário por parte da indústria.

3.5 – MENSAGEM \$GPRMC

No projeto proposto, será utilizada apenas a mensagem do tipo RMC (\$GPRMC). A sigla RMC é o acrônimo para Recommended Minimum sentence C (Mínima sentença em C recomendada). É a linha que atende as especificações do projeto e possui nos seus valores dados necessários como posicionamento (latitude e longitude), velocidade e tempo.

Observa-se também, que o módulo GPS/GSM adotado, possui a característica do envio desta mensagem \$GPRMC para o servidor da aplicação, utilizando o serviço GPRS. A tabela 4 explica cada informação integrante do código \$GPRMC enviado pelo módulo adotado ao projeto.

Exemplo de Mensagem \$GPRMC:

\$GPRMC,161229.487,A,25.0377,N,121.3366,E,0.13,309.62,120598,1.2,E,*10<CR><LF>

Tabela 4 - Significado dos campos da mensagem \$GPRMC

\$GPRMC,<1>,<2>,<3>,<4>,<5>,<6>,<7>,<8>,<9>,<10>,<11>,<12>*<13><14>				
Campo	Nome	Exemplo	Unidade	Descrição
1	ID da Mensagem	\$GPRMC		Cabeçalho do protocolo RMC.
2	Tempo UTC	161229.48 7		Horário UTC no formato: hh mm ss.sss.
3	Estado	A		A = Dados Válidos V = Dados Inválidos.
4	Latitude	3723.2475		Latitude formato: dd mm.mmmm.
5	Indicador de N/S	N		N = Norte e S = Sul.
6	Longitude	12158.341 6		Longitude no formato: ddd mm.mmmm.
7	Indicador de E/W	W		E = Leste e W = Oeste.
8	Velocidade	0.13	Nós	Velocidade dada em Nós.
9	Curso	309.62	Graus	Curso em graus.
10	Data	120598		Dados UTC de uma posição fixa no formato, dd mm yy.
11	Variação Magnética		Graus	Variação magnética em graus ou em direção (E = Leste e W = Oeste).
12	Modo	A		A = Modo Autônomo D = Modo Diferencial E = Modo Estimado M = Modo de Entrada Manual N = Dados Inválidos S = Modo de Simulação
*<13>	Checksum	*10		Código de erro da mensagem, caso o tenha ocorrido.
<14>	Fim de Mensagem	<CR><LF>		Indica do Fim da Mensagem.

Fonte: adaptado de NMEA, 2002.

3.6 – MÓDULO GPS TRACKER

Neste projeto, observasse a aplicação de um módulo (hardware) para obtenção de informações sobre o posicionamento do veículo além da data, velocidade e altitude no globo.

Definiu-se a utilização do aparelho GPS Tracker TK-102, fabricado pela empresa chinesa Xexun, ao projeto proposto. Este é um módulo que possui características de um receptor GPS integrado a um modem GSM portador do serviço GPRS para a transmissão de informações a um servidor remoto. Sua configuração se dá por meio de mensagens SMS que são enviadas ao módulo através de um aparelho celular, e o mesmo responde com um SMS de sucesso ou falha.



Figura 9 - GPS Tracker TK-102 – Fonte: Xexun, 2010

O módulo em seu funcionamento envia para um servidor previamente configurado, por intermédio da rede GSM e protocolo GPRS, uma mensagem de dados que contém informações como: latitude, longitude, velocidade, data, altitude, imei do chip GSM instalado

no módulo bem como outras que não serão utilizadas no projeto proposto. Através destas informações obtidas, será realizado o rastreamento do veículo que porta o módulo TK-102 em seu interior, adquirindo a característica de um módulo “embarcado”.

Na tabela são apresentadas as características detalhadas do módulo GPS.

Tabela 5 - Especificações do GPS Tracker TK-102

Conteúdo	Especificação
Dimensões	64mm x 46mm x 17mm
Peso	50g
Rede	GSM/GPRS
Banda	850/900/1800/1900Mhz ou 900/1800/1900Mhz
Chip GPS	SIRF3 chip
Módulo GSM/GPRS	Simcom300/Simcom340
Sensibilidade GPS	-159dBm
Exatidão GPS	5m
Tempo de início	Estado Frio 45s Estado Morno 35s Estado Quente 1s
Bateria	Chargeable changeable 3.7V 800mAh Li-ion battery
Standby	48 horas
Temperatura de armazenamento	-40°C até +85°C
Temperatura de Operação	-20°C até +55°C
Umidade	5%--95% sem condensação

Fonte: Xexun, 2010.

3.7 – LINGUAGENS PARA DESENVOLVIMENTO WEB

3.7.1 – LINGUAGENS DE MARCAÇÃO: HTML, XHTML e DHTML

HTML é uma linguagem de marcação para descrever páginas web. Seu nome é o acrônimo para Hyper Text Markup Language (Linguagem de marcação de hipertexto). Atualmente encontra-se na versão estável 4.01. Proposta em 1989 por Tim Berners-Lee, mesmo inventor da web, o HTML utiliza tags de marcação para descrever as páginas web. Podemos citar exemplos de tags como `<h1>` e `</h1>` que são utilizadas para formatar o texto em cabeçalhos, parágrafos, listas, links, etc.

Documento HTML é o mesmo que Página web. Os documentos HTML são descritivos de páginas web que contém tags HTML e textos simples. O objetivo de um web browser (como Internet Explorer ou Firefox) é a leitura de documentos HTML realizar sua exibição como página da web. O navegador não exibe as tags HTML, mas faz o uso das mesmas para interpretar o conteúdo da página [W3SCHOOLS, 2010]. A atual estrutura do HTML permite também a incorporação de formulários, imagens, vídeos e outros objetos como linguagens adicionais, por exemplo o Java Script.

O Extensible HyperText Markup Language (XHTML TM) representa a evolução do HTML com módulos que reproduzem, englobam e ampliam o HTML. O XHTML é a evolução do HTML baseada no padrão do XML ao invés de SGML tornando-o uma versão mais rígida e "limpa" do HTML, com regras mais específicas e detalhadas. XHTML é o sucessor do HTML, e uma série de especificações foram escritas para o XHTML [W3SCHOOLS, 2010]. Em 26 de janeiro de 2000, este padrão tornou-se uma recomendação da W3C, que é o órgão responsável por regulamentar os padrões da internet. Atualmente é um formato compatível com o HTML 4.01 e também com todos os web browsers, bem como browsers de celulares, mesmo estes não possuindo o mesmo poder de processamento de um PC comum.

O DHTML (Dynamic HTML) é a arte de combinar HTML, JavaScript, DOM e CSS [W3SCHOOLS, 2010]. Ele não é uma linguagem ou um padrão web propriamente dito. Ele utiliza os recursos destas linguagens para gerar páginas web dinâmicas e interativas.

Com a possibilidade de alteração dinâmica da renderização do conteúdo de um documento, o DHTML dá aos autores a habilidade de se criar documentos HTML visualmente notáveis, que interagem com o usuário sem o peso de confiar ao servidor da aplicação complicados ajustes de páginas HTML para conseguir efeitos especiais [Dicionário

da Internet, 2010]. Ainda, de acordo com a W3C, o Dynamic HTML é um termo usado por alguns fornecedores para descrever a combinação de HTML, folhas de estilo e scripts que permite que documentos sejam animados.

Utilizando o HTML com sua evolução XHTML e o DHTML, será possível realizar a manipulação das informações da página de maneira dinâmica, possibilitando a criação de uma interface mais intuitiva e interativa para o usuário do sistema de rastreamento.

3.7.2 – FOLHA DE ESTILO EM CASCATA - CSS

CSS é o acrônimo de Cascading Style Sheets (Folha de Estilo em Cascata). É uma linguagem para aplicação de estilo na apresentação documentos escritos em linguagens de marcação. Folha de estilo em cascata é um mecanismo simples para adicionar estilos (p.ex., fontes, cores, espaçamentos) aos documentos Web [W3C, 2010]. Seu principal benefício é prover a separação entre o formato e o conteúdo de um documento [DRUGOS, 2010].

Podemos citar o HTML e o XML como exemplos de linguagens onde o CSS pode ser aplicado. HTML marca e estrutura textos, cabeçalhos, parágrafos, links, botões, formulários, imagens e demais elementos da página e CSS define cores, posicionamento na tela, estilos de linhas, bordas e tudo o mais relacionado à apresentação [MAUJOR, 2009].

O CSS pode ser utilizado também para criação da visualização para os vários tipos de mídia existentes como impressão, web, celulares, etc. O código específico para cada tipo de mídia irá variar, já a estrutura do documento permanecerá estática sendo o CSS o responsável pela modificação da visualização de acordo com a mídia utilizada na visualização do conteúdo.

3.7.3 – JAVASCRIPT

JavaScript é uma linguagem de programação criada em 1995 por Brendan Eich, funcionário da Netscape. Em princípio a linguagem se chamava LiveScript, em alusão ao script que pode ser invocado no momento da execução de um evento no browser do cliente. Após o sucesso inicial do LiveScript, a Netscape recebeu uma grande colaboração da Sun, e a partir daí, adotou-se o nome JavaScript. A Microsoft também criou a sua versão para a linguagem, chamada de JScript. Posteriormente as implementações foram padronizadas com o nascimento da ECMAScript (European Computer Manufacturers Association).

JavaScript nada tem a ver com o Java, a única semelhança é relacionada a leve maneira de se codificar nas linguagens. A linguagem nasceu para que a interação com o usuário fosse mais rica. Concebida para a validação de formulários do lado do cliente. É uma linguagem interpretada pelo navegador, possibilitando assim, a manipulação de elementos da página sem que seja efetuada uma nova requisição ao servidor, economizando tempo de processamento e memória.

3.7.4 – PHP: PREPROCESSADOR DE HIPERTEXTO - VERSÃO 5 (PHP5)

O PHP (acrônimo recursivo de "PHP: Hypertext Preprocessor") é uma linguagem interpretada, essencialmente criada para desenvolvimento na Web. É uma linguagem executada no servidor que pode ser incorporada ao HTML. É uma linguagem que não possui definição de tipo das variáveis (fracamente tipada). A partir de sua terceira versão, contava com mínimos recursos de orientação a objetos, uma característica que ficou no passado devido a adoção da maioria dos recursos da POO em sua atual versão, 5.

Criada em 1994 por Rasmus Lerdorf, possuía o nome de Personal Homepage Tools. Concebido para ser utilizado no desenvolvimento de sua página pessoal. Mais tarde, em 1997 Zeev Suraski abraçou o projeto e lançou o PHP 3, que possuía alguns recursos da orientação a objetos. No ano de 1999 juntamente a Andi Gutmans escreveram a Zend Engine o "motor" do PHP 4 e fundaram a Zend Technologies, empresa responsável pela coordenação de desenvolvimento do PHP.

Na sua versão atual - PHP 5 - o conceito de orientação a objetos fica claramente adotado, tornando assim o PHP uma linguagem orientada a objetos. Com isso, a linguagem abriu concorrência a linguagens mais robustas, como o JAVA.

A utilização do PHP ultrapassa os 20 milhões de sites segundo pesquisas recentes. Portais mundialmente conhecidos como Yahoo!, Twitter, YouTube e FaceBook são exemplos de corporações que utilizam o PHP como principal tecnologia para desenvolvimento web.

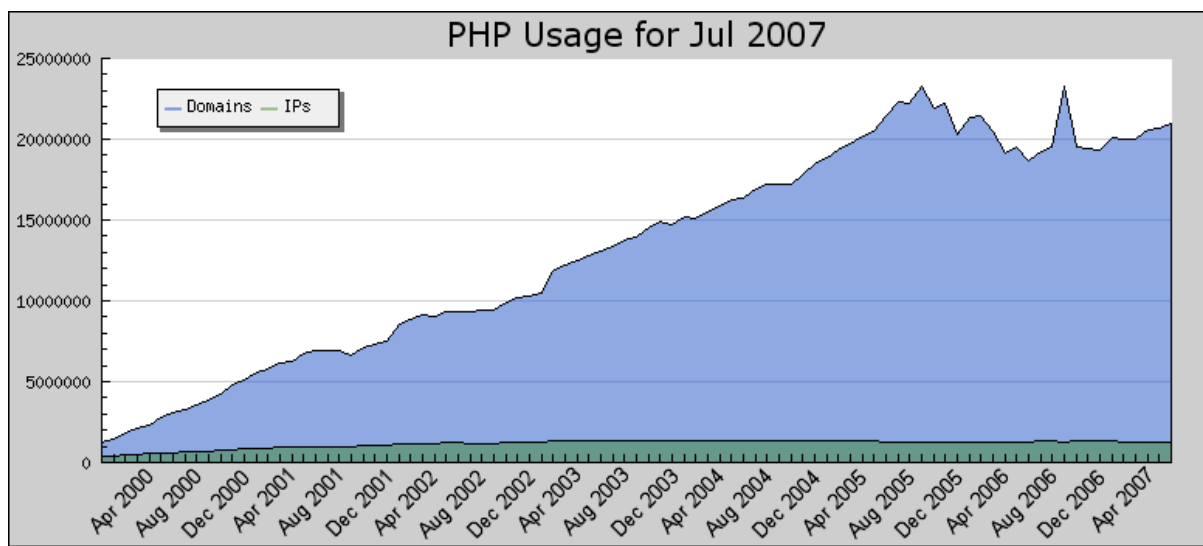


Figura 10 – Estatística de uso do PHP – Fonte: Netcraft, 2007.

A linguagem é a escolhida para o projeto, pois sua distribuição se dá sob licença GPL e também devido a sua característica de arquitetura multi-plataforma. É uma linguagem de código aberto que possui uma ampla comunidade distribuída pelo mundo. O custo com o desenvolvimento é muito baixo e a fonte para pesquisas é muito grande, além de sua curta curva de aprendizado.

3.8 – PROTOCOLOS E PADRÕES DE COMUNICAÇÃO

Telefonia móvel é a tecnologia utilizada para a comunicação por intermédio de voz, mensagem de texto e possui em sua arquitetura os protocolos de trocas de dados. Esta é uma ferramenta amplamente difundida e empregada para troca de informações durante o rastreamento do veículo. Utilizando o protocolo GPRS sob a tecnologia da rede GSM, as informações serão enviadas remotamente ao servidor onde se encontra a aplicação e onde se dá todo o processamento necessário.

3.8.1 – SISTEMA GLOBAL PARA COMUNICAÇÕES MÓVEIS - GSM

O Sistema Global para Comunicações Móveis - GSM - (Global System for Mobile Communications) é um padrão de tecnologia móvel e considerado o mais popular mundialmente. É um padrão aberto desenvolvido pela 3GPP (3rd Generation Partnership Project), uma associação que visa padronizar a criação, envio e reprodução de arquivos multimídia (vídeos) em telefones celulares e outros aparelhos wireless GSM.

Diante o crescimento da demanda por novas aplicações, novos serviços GSM são desenvolvidos. Nas 3 fases em que o GSM é dividido, podemos visualizar novos serviços como sendo marcos de criação de cada nova geração.

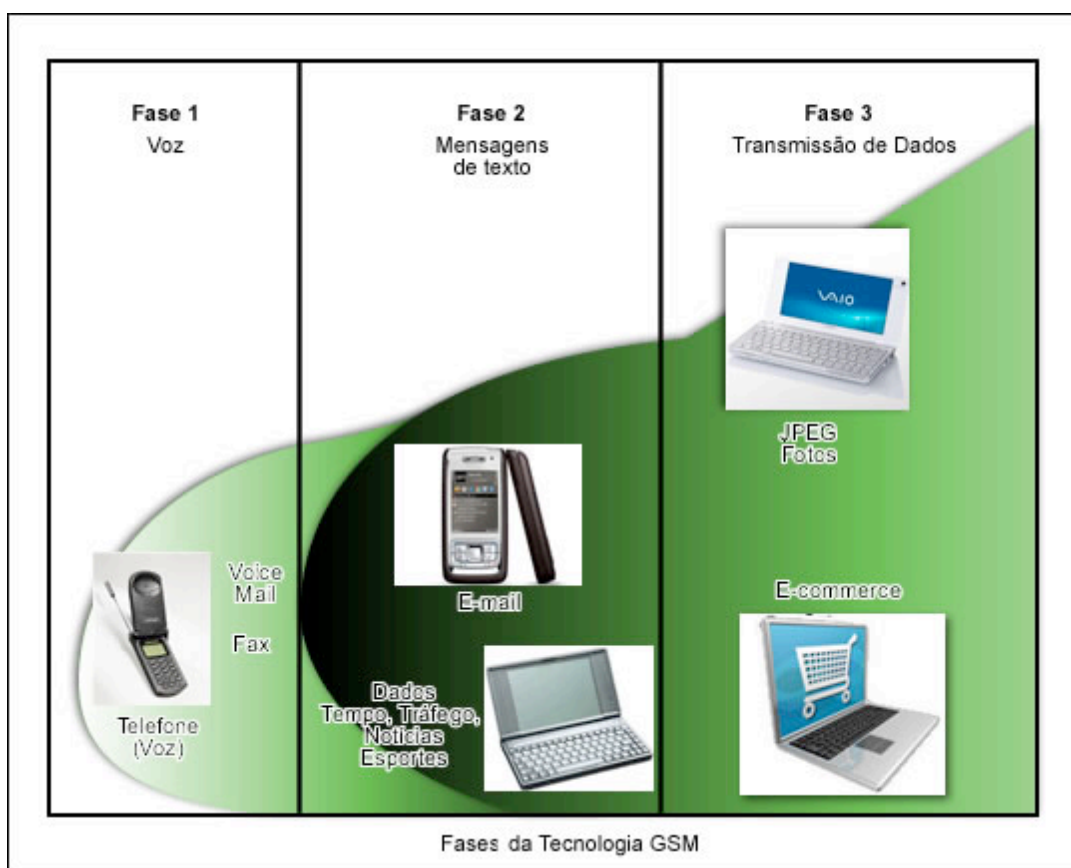


Figura 11 – Evolução do GSM – Fonte: Adaptado de [REDES GSM, GPRS, EDGE E UMTS, 2010].

As principais fases do GSM possuem características bem definidas. A fase 1 é marcada pela Telefonia, chamadas de emergência, SMS (mensagens curtas) ponto a ponto e multiponto, dados síncronos e assíncronos (taxas de transferência variando entre 0.3 e 9.6 kbps) e transmissão de pacotes assíncronos.

A fase 2 teve o surgimento dos serviços de e-mail, voz a meia taxa (half rate, serviço que permite ampliar o número de usuários, abrindo mão de certa parcela da qualidade da voz), melhorias no serviço de SMS, serviços de dados (como informações sobre tempo, clima, esportes entre outros), transmissão síncrona e dedicada de pacotes e a adição de serviços como identificador de chamadas, chamada restrita e teleconferência.

A fase 2+ do GSM é o marco necessário para que a integração do módulo GSM ao GPS se tornasse uma realidade e o projeto proposto pudesse ser executado. Esta fase é marcada pela introdução do serviço de dados por pacotes em altas taxas de transmissão, o GPRS.

3.8.2 – SERVIÇO DE RÁDIO DE PACOTE GERAL - GPRS

Com a revolução da internet, empresas prestadoras de serviços de telefonia celular, vislumbraram a possibilidade da utilização de suas redes como um novo meio para disseminação de informações provenientes da grande rede. Desde então, passaram a fornecer serviços de acesso à internet, através do celular.

A maior barreira a ser quebrada era a estrutura criada para oferecer a telefonia digital, que não estava preparada para o acesso à internet. A internet transporta dados por pacotes, através do protocolo IP e para que a rede móvel seja adaptada à internet, é preciso que os dados sejam organizados também em pacotes [Ricardo Di Lucia, 2008].

A partir daí, surgiu o GPRS acrônimo de General Packet Radio Services (Serviço de Rádio de Pacote Geral), protocolo desenvolvido para possibilitar o tráfego de dados por pacotes (Comutação por pacotes) na rede de tecnologia celular, provendo assim, serviços de internet nesta rede. A integração do GSM com o GPRS é considerada uma evolução essencial nas telecomunicações.

O GPRS é a tecnologia necessária para o aumento das taxas de transferência nas redes GSM existentes. As taxas de transferência do GPRS, em situações ideais poderá ultrapassar a marca dos 170kbps, porém a realidade é outra, já que na prática a taxa está em torno de 40kbps. No GPRS o serviço é "sempre ativo", ou seja, ele é um modo no qual os recursos somente são atribuídos a um usuário quando for necessário enviar ou receber dados. Esta técnica permite que vários usuários compartilhem os mesmos recursos, aumentando assim a capacidade da rede e permitindo uma gerência razoavelmente eficiente dos recursos. Isto permite às operadoras GPRS disponibilizar acesso à Internet móvel em alta velocidade e a um

custo razoável, pois a cobrança é feita pela quantidade de pacotes de dados transmitidos e não pelo tempo de conexão à rede [Consegurança, 2010].

Para o projeto proposto, adotou-se um módulo GPS que possui tecnologia GSM integrada e o serviço GPRS, onde por intermédio desta tecnologia é possível a transmissão das informações do veículo para um servidor que processa estas informações e faz o uso das mesmas. A escolha do mesmo se deu devido à utilização desta tecnologia que provê a mínima qualidade necessária aliada a um baixo custo.

3.9 – SERVIDORES E BANCO DE DADOS

A aplicação proposta será desenvolvida para web. Para tanto, se faz necessário um servidor de aplicação HTTP e um banco de dados para armazenar as informações obtidas através do programa escrito em PHP e também do módulo GPS/GSM/GPRS que será integrado ao sistema. Para suprir tal necessidade, foram escolhidos os servidores APACHE e o banco de dados relacional Postgresql.

3.9.1 – *SERVIDOR WEB APACHE*

O Projeto do Servidor Apache é uma tentativa de desenvolver e manter um projeto de servidor de código aberto para sistemas modernos, produzindo um servidor seguro, eficiente e expansível que esteja em sintonia com os padrões atuais. [APACHE FOUNDATION, 2010].

Apache, é o servidor para web mais utilizado no mundo. É considerado o servidor web livre mais bem sucedido do mundo, onde atualmente é utilizado por 57% do mercado de servidores web, dominando o mercado desde 1996. É a principal tecnologia da Apache Software Foundation, responsável por mais de uma dezena de projetos envolvendo tecnologias de transmissão via web, processamento de dados e execução de aplicativos distribuídos [APACHE FOUNDATION, 2010].

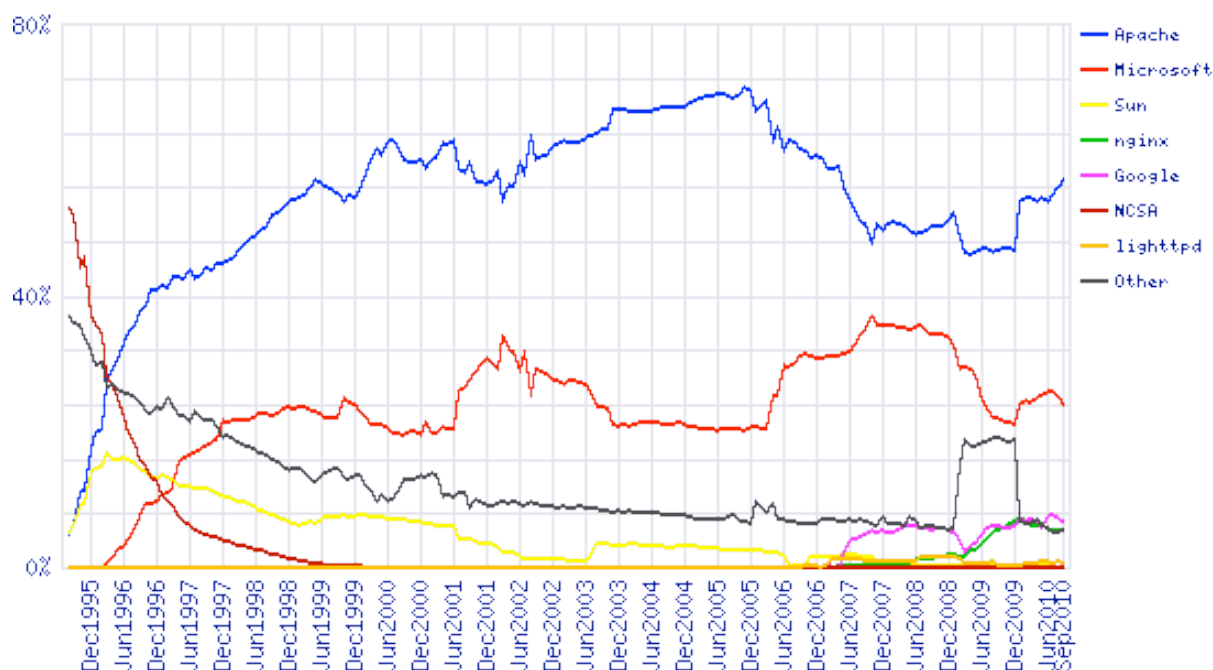


Figura 12 – Estatísticas de uso de softwares servidores WEB – Fonte: Netcraft, 2010.

É um servidor gratuito desenvolvido por membros da Apache Foundation. Um servidor web é a ferramenta de papel principal na atual arquitetura da web. É o responsável por receber as requisições advindas dos navegadores onde processa e devolve as informações processadas, de acordo como requisitado, ao navegador responsável pela requisição.

Suporta a versão 1.1 do protocolo HTTP. A sua estrutura, baseada em módulos, proporciona flexibilidade aos usuários que queiram escrever seus próprios módulos baseando-se na API do software. Com isso, suas funcionalidades podem ser mantidas ou também expandidas.

3.9.2 – BANCO DE DADOS POSTGRESQL

O PostgreSQL é um SGBD - Sistema de Gerenciamento de Banco de Dados - que possui características que atendem aos diversos padrões de aplicações, no âmbito da informática. É um SGBD relacional utilizado para armazenar informações de soluções de informática em todas as áreas de negócios existentes, bem como administrar o acesso a estas informações [ANDRÉ MILANI, 2008].

Após ter sua primeira versão estável lançada, em 1989, o banco alcançou tamanho sucesso que chamou a atenção da empresa Illustra Information Technologies, que em 1991 adquiriu o código do projeto, e após ter se fundido a Informix de proprietário o professor

Michael Stonebraker - co-fundador do projeto POSTGRES - foram adquiridas pela IBM em 2001.

Possui licença de uso BSD - Berkeley Software Distribution - que também é uma licença para softwares livres bastante flexível e possui diversas restrições a menos que a GPL - atual licença do linux. Mesmo após a aquisição do código por outra empresa, os mantenedores do projeto decidiram manter tudo relacionado a licença do software como estava.

O sistema criado, está em uma versão que é considerada estável e confiável. Segue o padrão ANSI SQL e disponibiliza todas as ferramentas que um SGBDOR pode disponibilizar. É um banco sem limite de armazenamento de dados, onde sua limitação se dá apenas no tamanho de suas tabelas, que é de no máximo 32TB por tabela. O hardware onde as informações estão armazenadas também pode ser considerado um limitador. Além do mais, é possível ter registros (linhas) com até 1.6TB, campos com até 1GB, tabelas com até 1.600 campos e índices ilimitados para aceleração da busca dos resultados [ANDRÉ MILANI, 2008].

É suportado pelos principais sistemas operacionais do mercado onde também foram disponibilizados drivers de conexão para as principais linguagens do mercado como C, C++, PHP, ASP, .NET, Ruby, TCL, Python, JSP e Java, entre outras. Provê recursos como: utilização de transações, replicação, clusterização, multithreads, segurança ssl, criptografia e uso de objetos GIS - com a extensão PostGIS -, entre outros.

O banco de dados PosgreSQL, escolhido para esta aplicação, possui flexibilidade de licença e também alta facilidade de implementação e manutenção dos dados do banco. É um software que atende às necessidades relacionadas ao armazenamento de dados do projeto proposto.

3.10 – BIBLIOTECAS UTILIZADAS

3.10.1 – *FRAMEWORK PHP: FRAMECALIXTO*

Para a criação de uma arquitetura funcional da aplicação proposta no projeto e a utilização da linguagem PHP 5 de uma maneira mais simplificada, adotou-se um framework. Segundo (Minetto, 2007), um framework de desenvolvimento é uma “base” de onde se pode desenvolver algo maior ou mais específico. É uma coleção de códigos-fonte, classes, funções, técnicas e metodologias que facilitam o desenvolvimento de novos softwares. Os frameworks facilitam o desenvolvimento de software, permitindo que os programadores se ocupem mais com os requerimentos do software do que com os detalhes tediosos, de baixo nível do sistema. Com o uso de frameworks, os programadores voltam a ter o controle de seu tempo e de seus códigos-fonte, as tarefas repetitivas são minimizadas, os projetos são concluídos em menos tempo, os padrões são seguidos, e a programação volta a ser uma arte [Minetto, 2007].

Para o projeto proposto, após várias pesquisas o framework de nome Framecalixto possuiu vários requisitos para que fosse proferida a sua escolha. Este é um framework em PHP desenvolvido para agilizar e simplificar a construção de sistemas WEB. O framework é baseado na estrutura MVC, onde é implementada a separação da camada de negócio e a camada de internacionalização, tornando-se assim um framework 'n' camadas. O framework possui também um gerador de cadastros, e a possibilidade de utilização de um controle de acesso e um controle de menus [Calixto, 2010].

A arquitetura do framework é baseada no padrão de projeto MVC, como mencionado anteriormente. MVC é um acrônimo para Model, View, Controller (Modelo, Visão e Controlador).

A ideia é separar todo o desenvolvimento de uma aplicação em três camadas.

- Model – gerencia o comportamento dos dados da aplicação.
- View – gerencia a saída gráfica e textual da parte da aplicação visível ao usuário.
- Controller – interpreta as entradas de mouse e teclado do usuário, comandando a Visão e o Modelo para se alterarem de forma apropriada.

O Controller é o responsável por receber as requisições efetuadas pelo usuário. Ele utiliza o model para manipulação destes dados e é de sua responsabilidade também a invocação da View correta, de acordo com a ação executada a partir dos resultados provenientes do Model. A maior vantagem na utilização deste padrão é a separação da camada lógica da camada de apresentação. Um designer pode trabalhar na camada de

apresentação definindo o HTML, CSS e etc. Por um outro lado, o administrador do banco de dados poderia trabalhar no modelo, enquanto o programador se concentraria nas regras de negócio pertencentes ao controlador. Desta forma, a mudança em uma das camadas por parte do seu responsável, teria pouco ou nenhum impacto nas demais camadas da aplicação.

3.10.2 – BIBLIOTECA JAVASCRIPT: JQUERY

Para facilitar a manipulação do HTML, a biblioteca escrita em JavaScript jQuery é aplicada ao projeto. JQuery é uma biblioteca JavaScript que de maneira rápida e concisa simplifica a maneira de percorrer a estrutura de documentos HTML, manipulação de eventos, animação e interações Ajax para agilizar o desenvolvimento web. É uma biblioteca projetada para alterar o conceito da escrita do JavaScript [jquery.com, 2010].

3.10.3 – BIBLIOTECA DE MAPAS SOB DEMANDA: GOOGLE MAPS API

A API JavaScript Google Maps, permite que sejam utilizados os mapas fornecidos sob demanda pela empresa Google no sistema proposto. Os mapas poderão ser incorporados e manipulados através desta biblioteca. Atualmente em sua versão 3, a biblioteca possui uma nova codificação que apresenta mais estabilidade, velocidade, maior compatibilidade com dispositivos móveis e com os principais navegadores web disponíveis no mercado. Esta API fornece várias ferramentas para que os mapas que serão utilizados no sistema possam ser manipulado de maneira simples, bem como utilitários que serão necessários para maior interação com o usuário do sistema.

A Google Maps JavaScript API V3 é um serviço gratuito, disponível para qualquer site que o público possa usar gratuitamente [Google, 2010]. Para uma aplicação paga, será necessário acrescentar ao valor dos custos, o valor da aquisição deste serviço de forma comercial.

Atualmente, a API V3 conta com um recurso que visa solucionar o "Problema do Caixeiro Viajante" PCV (TSP - Travelling Salesman Problem em inglês), para os mapas que serão utilizados no projeto.

O PCV pode ser definido como o problema de encontrar o percurso de menor distância ou custo, que passa por um conjunto de cidades sendo cada cidade visitada exatamente uma vez [Cunha, 2002].

O PCV é um problema de otimização que, apesar de parecer modesto é, na realidade, muito investigado por cientistas, matemáticos e investigadores de diversas áreas, tais como: logística, genética e produção, entre outros [Applegate et al., cop. 2006, p. 1]. O problema pertence à categoria NP-difícil, NP-hard (em inglês) , que o remete para um campo de complexidade exponencial, isto é, o esforço computacional necessário para a sua resolução cresce exponencialmente com o tamanho do problema. Assim, dado que é difícil, se não impossível, determinar a melhor solução desta classe de problemas, os métodos de resolução passam pelas heurísticas e afins que, do ponto de vista matemático, não asseguram a obtenção de uma melhor solução (Cunha, 2002).

Para a tarefa de sugestão do melhor percurso a ser seguido pelo veículo no momento do transporte da mercadoria, o recurso da biblioteca aplicado mostrou-se eficiente. O recurso atualmente possui a característica de sugerir percursos para até 9 pontos de entrega com possível melhoria de maior quantidade de pontos no futuro.

CAPÍTULO 4 – MODELO PROPOSTO

Após a definição da arquitetura a ser utilizada, escolha do hardware a ser integrado ao projeto e a apresentação do funcionamento de cada uma destas tecnologias, a fase do desenvolvimento é iniciada. Se faz necessário que as etapas previstas no planejamento sejam concluídas uma a uma.

No decorrer deste capítulo são expostas as etapas do desenvolvimento até se chegar ao produto final deste projeto: o sistema para rastreamento de frotas online denominado de *iGeoRastreador*.

4.1 – APRESENTAÇÃO GERAL DO MODELO PROPOSTO

O protótipo do projeto faz uso de um módulo GPS integrado a um modem GSM que utiliza o serviço GPRS para comunicação com o servidor da aplicação, como é um módulo de rastreamento simples, que pode ser utilizado até mesmo para rastrear uma pessoa, não há necessidade de uma instalação diferenciada em um veículo, basta que o mesmo esteja em seu interior, de preferência em um local de difícil acesso.

O módulo executa a tarefa de envio de uma mensagem no padrão NMEA ao servidor que possui o script PHP denominado de "Importador de Dados", responsável pelo tratamento desta mensagem. Após a aquisição e tratamento da mensagem, as informações são divididas em blocos e inseridas na tabela do banco de dados.

Uma rotina implementada no programa da central de monitoramento é a responsável pela verificação da existência de registros na tabela de rastreamento. Estes registros são convertidos em informações que serão inseridas em outra tabela (uma forma de manter um histórico), e utilizados pela central de monitoramento para exibição do posicionamento do veículo no mapa.

Para um sistema de rastreamento de veículos, é necessário que se tenha uma base de dados de informações previamente cadastradas, no momento da implantação do sistema, para que então rotinas implementadas efetuem o tratamento de dados e gere resultados esperados.

Informações provenientes do GPS são confrontadas com informações previamente cadastradas nos cadastros auxiliares gerados no processo de implementação deste projeto.

São cadastros auxiliares ao sistema de rastreamento de veículos:

- Cadastro de Empresas
- Cadastro de Funcionários
- Cadastro de Veículos
- Cadastro de Detalhes dos Veículos
- Cadastro de Ocorrências
- Cadastro de Locais
- Cadastro de Tipos de Locais
- Cadastro de Percursos (engloba o cadastro de trechos de percursos)
- Cadastro de Clientes
- Cadastro de Encomendas

Todos os cadastros auxiliares foram criados de forma sucinta para que se tornasse viável a demonstração deste protótipo. Caso se torne um sistema comercial, maiores especificidades podem ser acrescentadas mediante a um levantamento de requisitos aprofundado em uma empresa do ramo de transportes.

A central de monitoramento pode ser vista como o "cérebro" do sistema, e todos os outros cadastros auxiliares são levados em consideração para tornar esta central funcional.

Foram implementadas duas modalidades para execução do rastreamento: Cumprimento de percursos previamente cadastrados para entrega de encomendas ou rastreamento livre, onde o veículo será acompanhado na central de monitoramento por um funcionário da empresa, apenas com o intuito de se ter controle sobre o deslocamento do mesmo.

4.1.1 – ESTIMATIVA DE CUSTOS

Abaixo, a tabela com informações sobre todos os custos acumulados do projeto.

Tabela 6 - Estimativa de Custos do Projeto

Equipamento	Fonte	Custo
Módulo GPS Tracker TK-102	Mercado Livre	R\$350,00
Licença do Apache	Apache V2	Gratuita
Licença do PHP	PHP License	Gratuita
Licença do PostgreSQL	GPL	Gratuita
Licença da API GoogleMaps	-	Gratuita
Plano de Internet 3G	Vivo	R\$89,90/mês
Hospedagem Servidor Web	BlueHost	R\$7,50/mês
Aquisição de IP Fixo	BlueHost	R\$4,50/mês
Total		R\$451,90

O custo mensal de aquisição da hospedagem para o sistema é de R\$12,00, dependente da provedora de serviços escolhida, no caso a BlueHost.

O gasto com aquisição do módulo GPS (um por veículo) é R\$350,00, visto que para a utilização do aparelho não há necessidade do pagamento de mensalidade.

Outro custo calculado mensalmente é o da contratação do serviço de dados 3G junto a uma operadora de telefonia. É necessária uma assinatura de plano de dados por veículo, sendo que um possível acordo com a operadora (devido à quantidade de assinaturas) poderia reduzir este custo.

4.2 – DESCRIÇÃO DAS ETAPAS DO MODELO

4.2.1 – INSTALAÇÃO E CONFIGURAÇÃO DO MÓDULO GPS

O módulo GPS TK-102 da fabricante Xexun escolhido para o projeto, não requer instalação de forma específica no veículo, o qual deverá fornecer informações para que se torne possível realizar o rastreamento. Como é um módulo simples, basta que o mesmo esteja no interior do veículo, dotado de um chip GSM com uma conta que possua pacote de dados ativo. Para que se tornasse uma ferramenta útil ao projeto e para que não existisse o risco de fraudes no aparelho, o mesmo teve de ser instalado em um local de difícil acesso à pessoas não autorizadas a manipulá-lo.

A bateria do aparelho obteve um bom desempenho nos testes e mesmo sua especificação informando que o tempo de espera é de 48 horas, o aparelho demonstrou uma durabilidade bastante superior e funcionou durante 60 horas, não sendo assim necessário a este protótipo a aquisição do cabo de ligação do aparelho diretamente à bateria do veículo, visto que a bateria dura um tempo aceitável para a operação de rastreamento, fazendo com que os custos fossem reduzidos.

Para comunicação entre o módulo e o servidor, observa-se o uso de um chip GSM, o mesmo utilizado em celulares, cujo assinatura e ativação do pacote de dados foram realizadas juntamente a operadora VIVO. Abaixo, a imagem do módulo GPS/GSM/GPRS já com o CHIP adquirido na operadora, instalado. O número do chip utilizado para o módulo é: (61) 9952-2241.



Figura 13 – Módulo TK-102 com o CHIP GSM instalado.

Com os passos anteriores concluídos, é necessário que se ligue o aparelho. Após pressionar o botão On/Off (localizado na lateral do aparelho), o módulo iniciará o funcionamento, sinais GSM serão adquiridos e a triangulação com os satélites GPS será realizada. O tempo para que os sinais sejam estabilizados está entre 10 e 40 segundos, tendo um tempo maior de espera caso os fatores de erro do GPS estejam influenciando nesta aquisição de sinais.

Após instalação do CHIP GSM no módulo e sua acomodação no veículo de maneira segura, inicia-se a fase de configuração do mesmo. São enviados comandos de configuração fornecidos no manual do módulo, via mensagens de texto curtas (SMS) de um aparelho celular comum.

A unidade é reinicializada em seu primeiro uso, após o envio da mensagem "begin+SENHA", sendo a senha padrão: 123456.



Figura 14 – Primeira mensagem de configuração do módulo GSM/GPRS/GPS

O próximo passo é configurar o tempo em que as mensagens serão enviadas ao servidor. O comando utilizado é: "t030s100n123456". A primeira parte do comando (t030s) diz respeito ao tempo de envio das mensagens, definiu-se o tempo de 30 segundos. A segunda parte (100n) diz respeito à quantidade de mensagens enviadas, o máximo aceito pelo módulo é 255 mensagens, porém para fins de testes, definiu-se o envio de 100 mensagens sendo uma mensagem enviada a cada 30 segundos.



Figura 15 - Segunda mensagem de configuração do módulo GSM/GPRS/GPS

Para definição de qual IP as mensagens serão enviadas, utiliza-se o comando: "adminip12345 IP PORTA". Juntamente à provedora BlueHost, contratou-se a conta para hospedagem do sistema (disponível online em: www.ueiner.com/igeorastreador). Também foi necessária a aquisição de um IP dedicado. A provedora forneceu o IP: "69.195.78.117" e a porta "1234" teve de ser liberada no firewall via atendimento da empresa. Após as solicitações atendidas, o módulo pôde ser configurado com o comando: "adminip12345 69.195.78.117 1234".

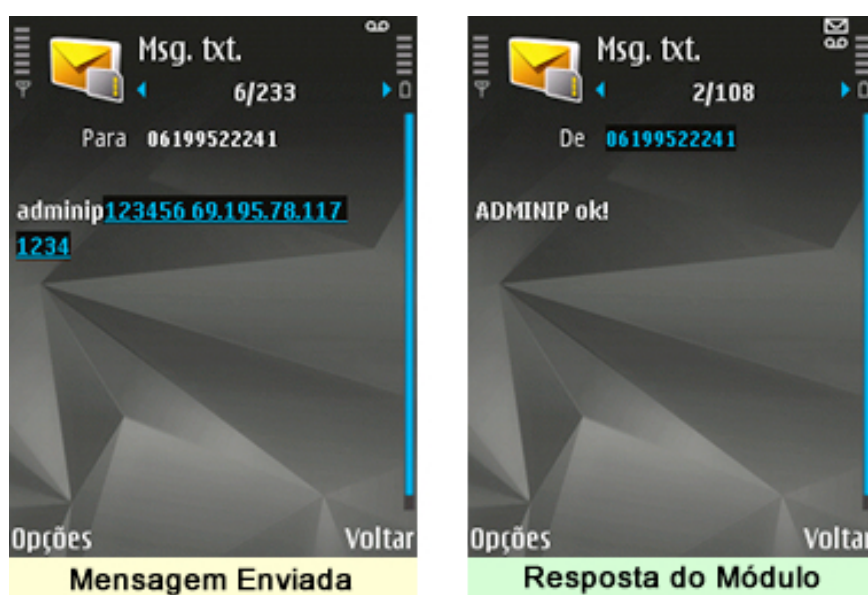


Figura 16 - Terceira mensagem de configuração do módulo GSM/GPRS/GPS

A portadora de dados utilizada para o envio das mensagens com o serviço GPRS é definida logo em seguida. A contratação do pacote de dados se deu juntamente à operadora VIVO, que possui a portadora de dados com endereço "zap.vivo.com.br", usuário "vivo" e senha "vivo".

A Figura 17 ilustra como se realizou a configuração.

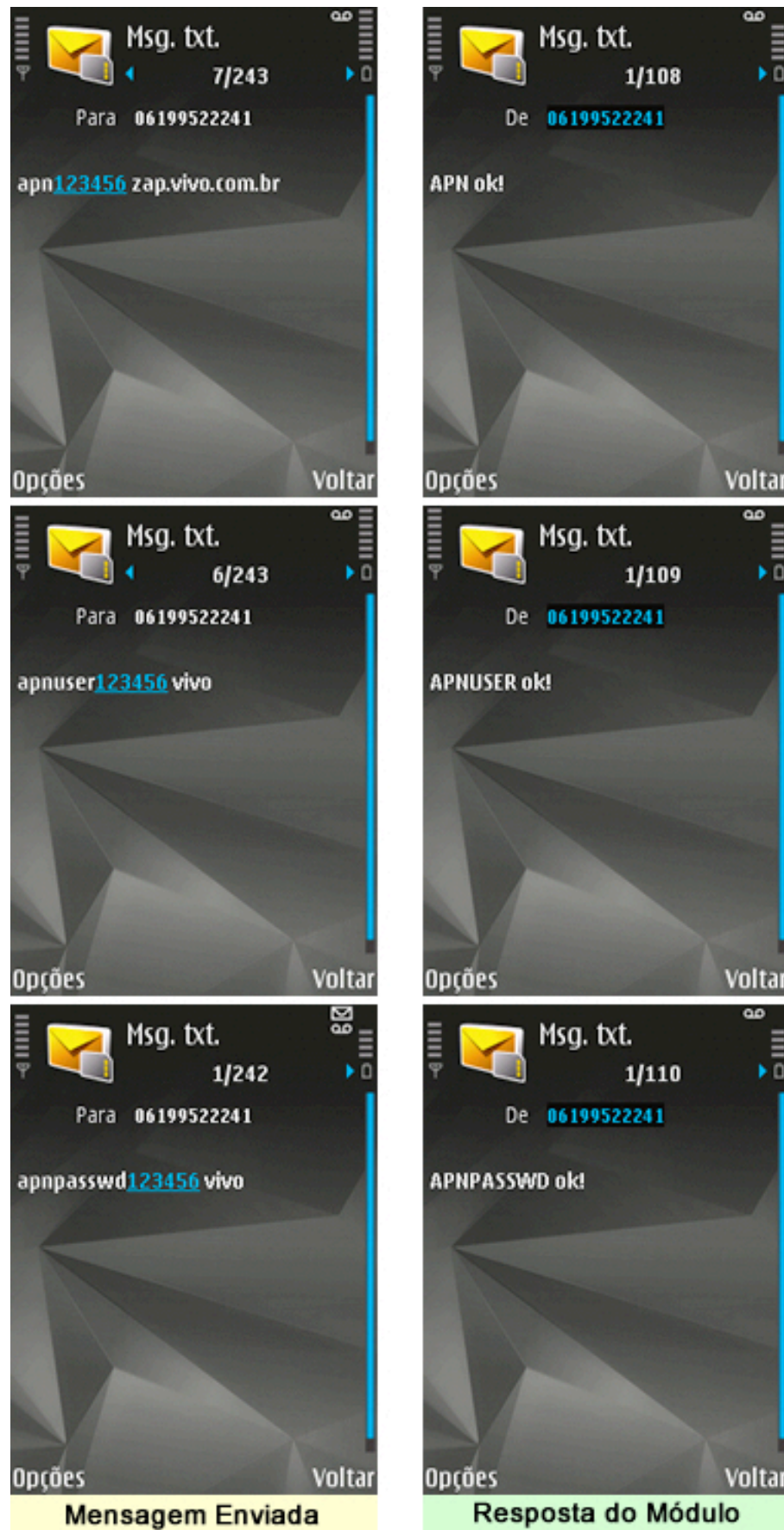


Figura 17 – Configuração para a portadora de dados VIVO no módulo GSM/GPRS/GPS.

4.2.2 – TRATAMENTO DAS MENSAGENS PROVENIENTES DO MÓDULO GPS

O hardware escolhido para coleta de informações, é o TK-102 da fabricante Xexun. O módulo envia as informações a um servidor web via TCP sob o serviço GPRS da operadora, através de um IP e uma porta liberada no firewall deste servidor.

Este pacote de informações contém uma mensagem no formato NMEA (string no padrão \$GPRMC) acrescida de outras informações do módulo TK-102.

```
Chamada da função x no arquivo:/Users/ueiner/Sites/projetoFinal/igeorastreador/testeImportador.php na linha:11
(string) = "101017193522,06184042241,GPRMC,193522.000,A,1549.8564,S,04803.7881,W,0.00,36.76,171010,,,A*52,F,,
imei:011412001070485,05,1209.7,F:3.92V,0,139,16845,,,01CD,0855"
```

Figura 18 – Exemplo de mensagem enviada pelo módulo TK-102

Para que se tornasse possível a captura da mensagem do lado do servidor web, se fez necessário o desenvolvimento de um script PHP que permanecesse ativo "escutando" (*listener*) todas as informações que fossem encaminhadas pelo módulo TK-102 à porta configurada. Para o iGeoRastreador escolhemos a porta "1234".

O php possui funções específicas para acesso a informações em portas da rede, para tanto utiliza-se de soquetes de rede para estabelecer tais conexões.

Conforme configurado, a cada 30 segundos uma mensagem será enviada pelo módulo, o script estará sempre ativo aguardando a chegada destas informações que serão tratadas pelo mesmo e inseridas no banco. O nome dado ao script é *importador de dados*.

O script PHP importador de dados, faz o papel de "escutador" de informações. Estas informações recebidas são analisadas, tratadas e dependendo da sua importância, devido a certos parâmetros, serão inseridas na tabela do banco de dados modelada para receber as informações provenientes do rastreamento.

O script recebe a mensagem e a quebra por vírgulas, que é o seu caractere separador definido pela NMEA. Neste momento é utilizado o recurso de matriz da linguagem, o array.

O recurso array do php nada mais é que um mapa ordenado, otimizado para que possua índices associativos. É o mesmo recurso conhecido como matriz em outras linguagens, porem agrega a funcionalidade de se associar índices e valores, sendo os índices nomeados por quem os criou.

A mensagem recebida e quebrada é mapeada no array definido previamente, conforme observado na imagem 19.

```
(array) #27 índices:
[nr_serie]=> (string) = "101017193522"
[telefone_autorizado]=> (string) = "06184042241"
[tipo_mensagem]=> (string) = "GPRMC"
[tempo_utc]=> (string) = "193522.000"
[estado]=> (string) = "A"
[latitude]=> (string) = "1549.8564"
[latitude_tipo]=> (string) = "S"
[longitude]=> (string) = "04803.7881"
[longitude_tipo]=> (string) = "W"
[velocidade_nos]=> (string) = "0.00"
[curso]=> (string) = "36.76"
[data_ddmmyy]=> (string) = "171010"
[variacaomagnetica]=> (string) = ""
[modo_gps]=> (string) = ""
[checksum]=> (string) = "A*52"
[indicador_sinal_gps]=> (string) = "F"
[imei]=> (string) = "011412001070485"
[numero_gps]=> (string) = "05"
[altitude]=> (string) = "1209.7"
[forca_bateria]=> (string) = "F:3.92V"
[carregando]=> (string) = "0"
[tamanho_string]=> (string) = "139"
[crc_checksum]=> (string) = "16845"
[codigo_pais]=> (string) = ""
[codigo_rede]=> (string) = ""
[lac]=> (string) = "01CD"
[cell_id]=> (string) = "0855"
```

Figura 19 – Array mapeado com as informações provenientes do módulo TK-102

A cada envio de informação por parte do módulo TK-102, todas estas informações serão tratadas e inseridas no banco de dados para posterior utilização da central de monitoramento.

O código do importador de dados está inserido no apêndice deste documento.

4.2.3 – ESTRUTURA DE PASTAS DO SISTEMA

Para o projeto proposto, adotou-se o framework PHP FrameCalixto. O objetivo era facilitar o desenvolvimento dos cadastros auxiliares à central de monitoramento, onde dados relativos à empresa, veículos e percursos foram cadastrados.

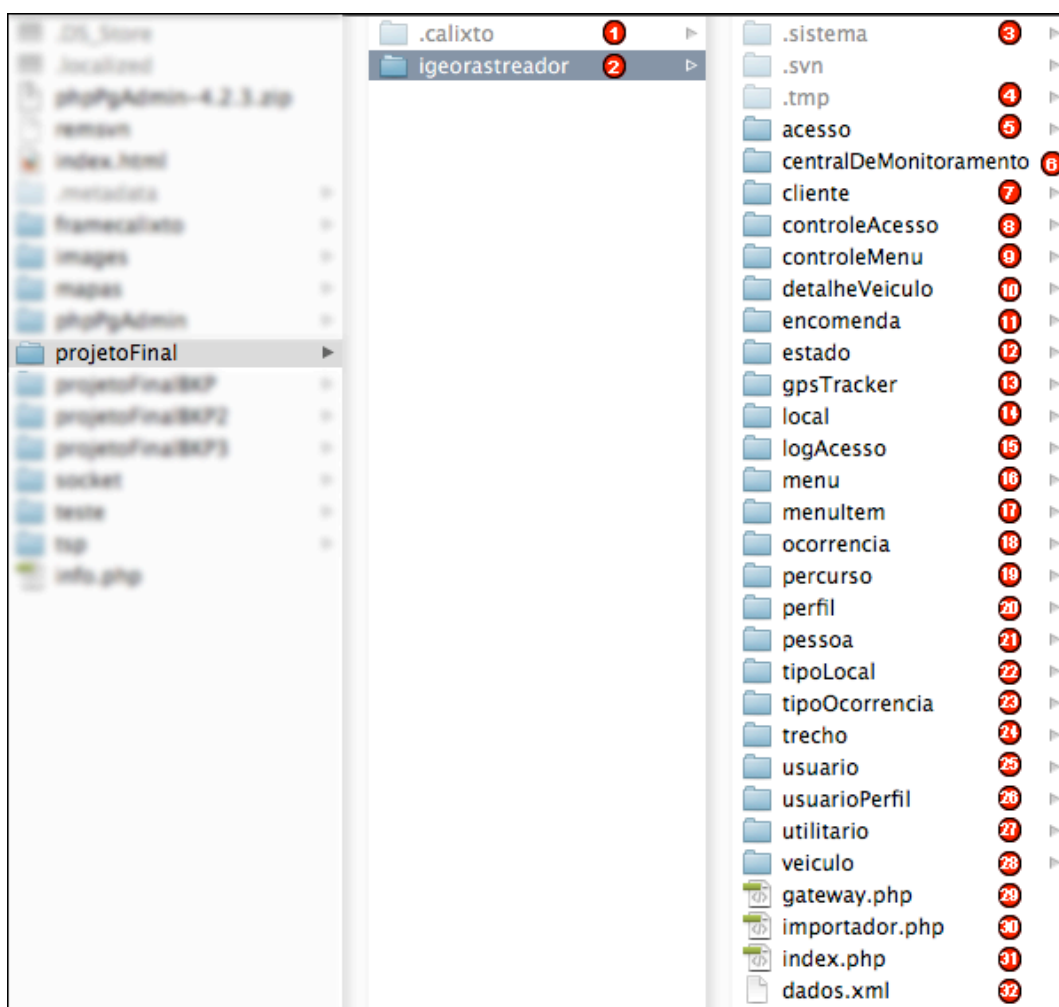


Figura 20 – Estrutura de pastas do sistema

- (1) – Pasta dos códigos do framework FrameCalixto.
- (2) – Pasta do projeto iGeoRastreador.
- (3) – Pasta onde ficam as configurações do sistema além configurações específicas de temas e imagens.
- (4) – Pasta para os arquivos temporários do sistema (cache de templates, que são os arquivos de visualização).

(5) a (28) – Pastas das entidades do sistema, as pastas contém os arquivos necessários para o funcionamento dos cadastros gerados para o sistema. Estas pastas estão subdivididas em 3 pastas: classes, html e xml.

A pasta classes contém arquivos de Controle (iniciados com a letra C maiúscula), arquivos de Negócio (iniciados com a letra N maiúscula), arquivos de Persistência (iniciados com a letra P maiúscula) e arquivos de internacionalização (iniciados com a letra I maiúscula).

A pasta html contém os arquivos de templates (visualização) mais específicos. Apenas templates que exibem informações de mapas foram alterados neste projeto.

A pasta xml contém arquivos de configuração da entidade, bem como arquivos para internacionalização dos formulários gerados para cadastro de informações.

(29) – Arquivo de índice para o funcionamento do sistema.

(30) – Importador de dados utilizado para análise e inserção no banco de dados, os dados provenientes do módulo TK-102. O arquivo terá sua execução inicializada através do terminal do sistema operacional utilizado pela empresa.

(31) – Arquivo responsável por receber todas as requisições do navegador web.

(32) – Arquivo com dados previamente cadastrados para que sejam importados na primeira execução do sistema.

4.2.4 – MODELO DE DADOS DO SISTEMA

Após o levantamento dos requisitos de um sistema de rastreamento de veículos, o modelo de dados do sistema pôde ser definido. Para exemplificar este modelo, gerou-se um MER para melhor visualização dos relacionamentos estabelecidos para gerenciamento do grande volume de dados gerados pelo sistema.

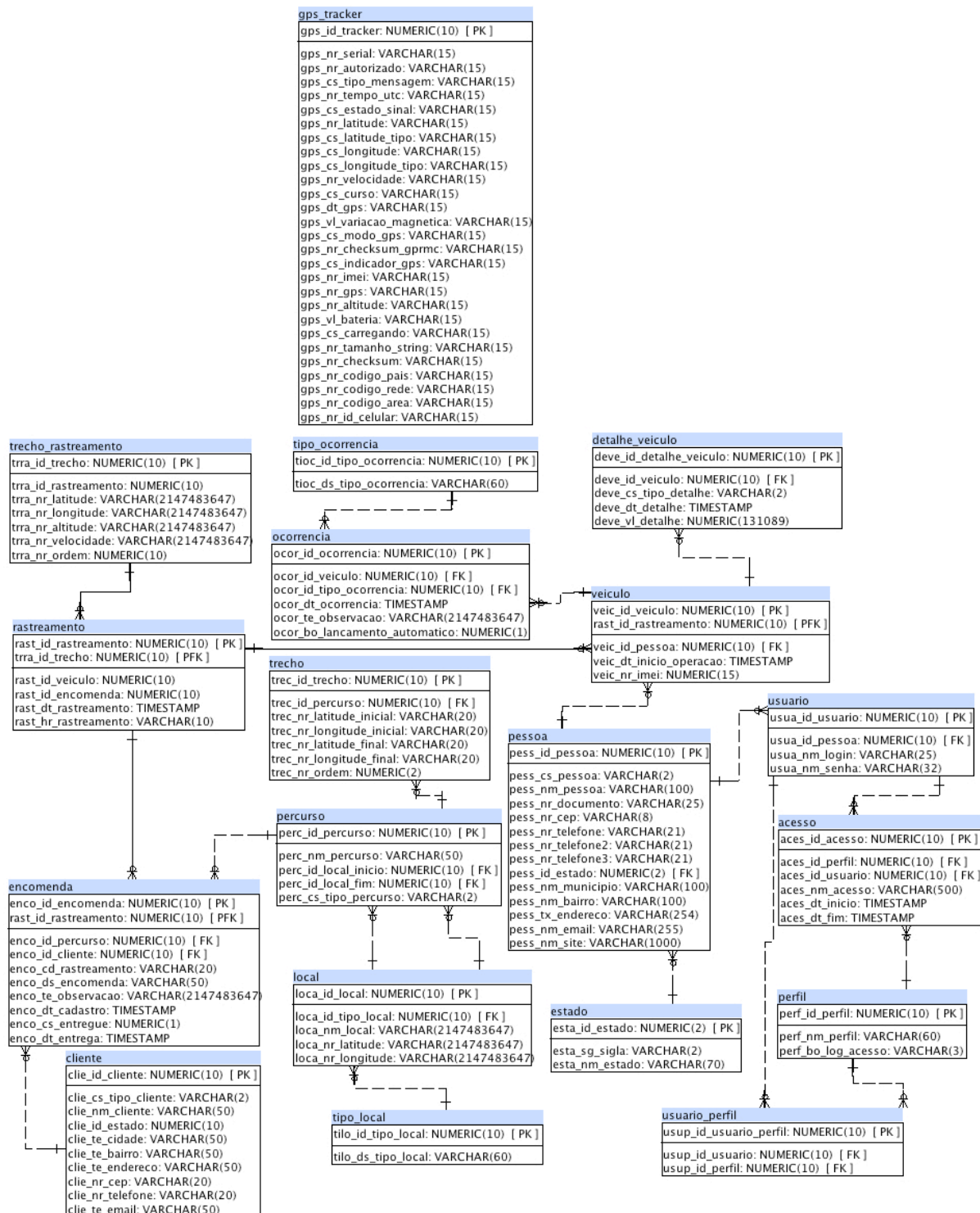


Figura 21 - Modelo Entidade Relacionamento do projeto

A tabela principal do sistema é a tabela *gps_tracker*, que armazena informações provenientes do módulo TK-102. Estas informações são analisadas pelo script da central de monitoramento, que faz o uso das mesmas para exibição nos mapas interativos e posteriormente as registra na tabela *rastreamento* para a manutenção de um histórico operacional.

As demais tabelas, são necessárias ao sistema para cadastramento de informações sobre a empresa que está utilizando o sistema. São as tabelas dos cadastros auxiliares.

4.2.5 – FUNCIONALIDADES DO SISTEMA

Para que se tornasse possível a implementação de um sistema de rastreamento de veículos cadastros auxiliares foram implementados com o intuito de fornecer informações previamente cadastradas, ao sistema. Para tanto, considerou-se que para este protótipo, dois perfis de atores eram necessários para manutenção das informações do sistema.

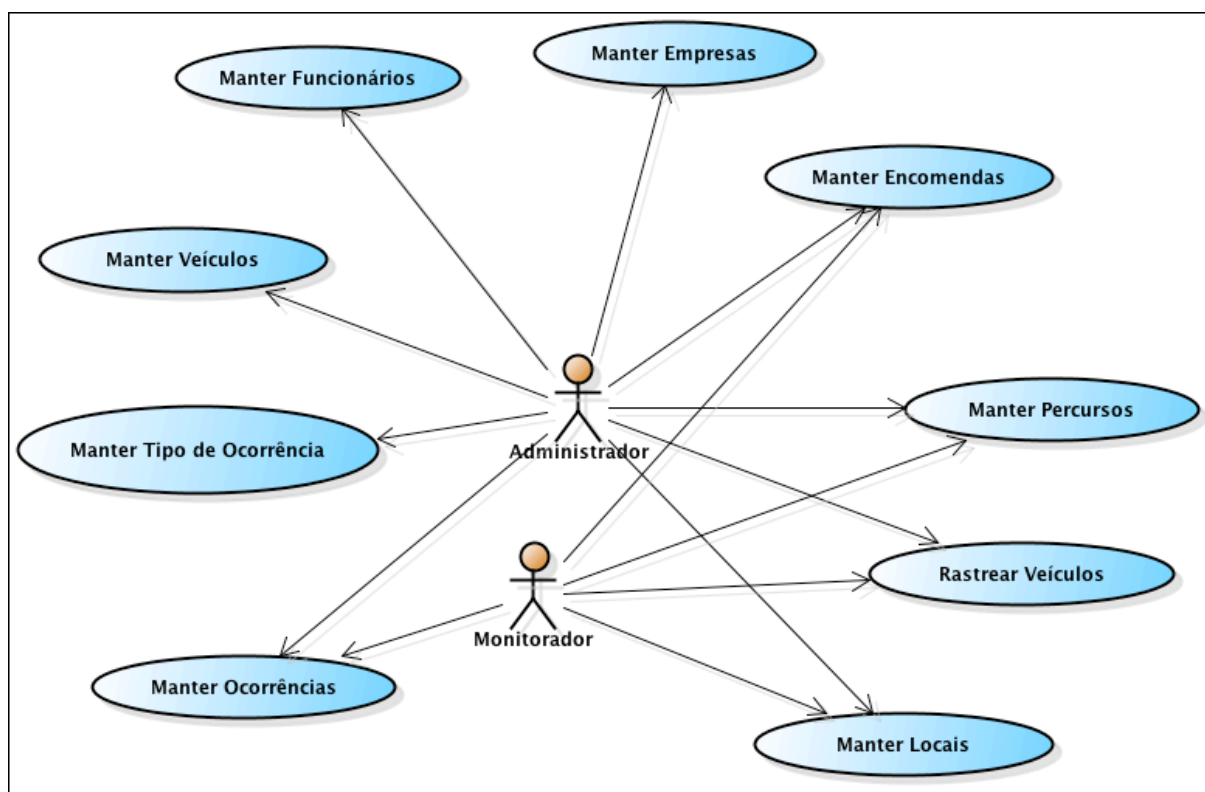


Figura 22 – Caso de uso principal do sistema, atores: Administrador e Monitorador.

O ator com perfil de administrador, é o responsável pela manutenção das informações de todos os cadastros do sistema. Possui acesso a todos os cadastros sem restrição. É o gerente da empresa. Já o ator com perfil de monitorador, é o encarregado da operação. Possui acesso

apenas aos cadastros menos sensíveis do sistema. Responsável pelo cadastramento das encomendas, manutenção das ocorrências, percursos, locais e por fim o responsável pelo rastreamento do veículo em si, acompanhando-o por intermédio da central de monitoramento.



Figura 23 – Menu de acesso aos cadastros, para o usuário com perfil de monitorador.



Figura 24 – Menu de acesso aos cadastros, para o usuário com perfil de monitorador.

As imagens 23 e 24 dizem respeito ao menu principal do sistema. O perfil de administrador possui livre acesso a todas as funcionalidades do sistema, conforme visto na imagem 23. Algumas funcionalidades não são exibidas para o usuário com perfil de monitorador, como pode ser visto na figura 24.

The screenshot displays the iGeoRASTREADOR web application interface. At the top, the logo 'iGeoRASTREADOR' is visible with the subtitle 'Sistema de Monitoramento de Veículos'. Below the logo is a navigation bar with icons and labels for 'PRINCIPAL', 'CADASTROS', 'ENCOMENDAS', 'CENTRAL DE MONITORAMENTO', 'APOIO', and 'SAIR'. A yellow message box at the top of the main content area contains the text: 'Preencha os dados e clique em gravar.' Below this is a form titled 'Pessoas - Edição'. The form contains the following fields: 'Tipo:' (a dropdown menu with a blue arrow icon and an asterisk), 'Nome:' (a text input field with an asterisk), 'CPF/CNPJ:' (a text input field), 'Telefone:' (a text input field), 'Celular:' (a text input field), 'Telefone Aux.:' (a text input field), 'CEP:' (a text input field), 'Estado:' (a dropdown menu with a blue arrow icon), 'Município:' (a text input field), 'Bairro:' (a text input field), 'Endereço:' (a text input field), 'E-mail:' (a text input field), and 'Site:' (a text input field). At the bottom of the form are two buttons: 'SALVAR' (with a floppy disk icon) and 'TELA DE PESQUISA' (with a magnifying glass icon).

Figura 25 – Formulário de Cadastro de Pessoas

O cadastro de pessoas engloba cadastro de pessoa de acordo com os tipos: física e jurídica. Para um melhor aproveitamento do cadastro, também foram acrescentados os tipos de pessoas físicas e jurídicas internas e externas. Pessoas físicas e jurídicas internas, dizem respeito à empresa e os funcionários da empresa. Pessoas físicas e jurídicas externas, dizem respeito a clientes da empresa.

The screenshot displays the iGeoRASTREADOR web application interface. At the top, the logo 'iGeoRASTREADOR' is visible, followed by the text 'Sistema de Monitoramento de Veículos'. Below the logo is a navigation bar with the following menu items: PRINCIPAL, CADASTROS, ENCOMENDAS, CENTRAL DE MONITORAMENTO, APOIO, and SAIR. A yellow banner below the navigation bar contains the instruction: 'Preencha os dados e clique em gravar.' Below this banner is a form titled 'Cadastro de Veículo - Edição'. The form contains the following fields: 'Empresa:' with a dropdown menu, 'Data de Início da Operação:' with a date picker, 'Número do Imei:' with a text input field, 'Placa:' with a text input field, and 'Descrição:' with a larger text input field. At the bottom of the form are two buttons: 'SALVAR' and 'TELA DE PESQUISA'.

Figura 26 – Formulário de Cadastro de Veículos

No cadastro dos veículos, se faz necessário o registro do número IMEI do dispositivo rastreador instalado no veículo. Este número é único e a partir dele será possível identificar qual módulo está enviando a informação do rastreamento para a central, identificando assim o veículo.

The screenshot displays the iGeoRASTREADOR web application interface, showing a different form titled 'Cadastro de Detalhe Veículo - Edição'. The navigation bar and yellow banner are identical to the previous screenshot. The form contains the following fields: 'Veículo:' with a dropdown menu, 'Tipo de Detalhe:' with a dropdown menu, 'Data de Cadastro:' with a date picker, and 'Valor:' with a text input field. At the bottom of the form are two buttons: 'SALVAR' and 'TELA DE PESQUISA'.

Figura 27 - Formulário de Cadastro dos Detalhes do Veículo

O cadastro de detalhes do veículo é o responsável por manter um histórico de algumas informações do veículo. A quilometragem inicial e acumulada do veículo são controladas através deste cadastro. Esta informação é atualizada automaticamente por intermédio da central de monitoramento, após a análise das informações provenientes do rastreamento. A quilometragem dos pneus do veículo também poderá ser controlada neste cadastro e atualizada pela central de monitoramento.

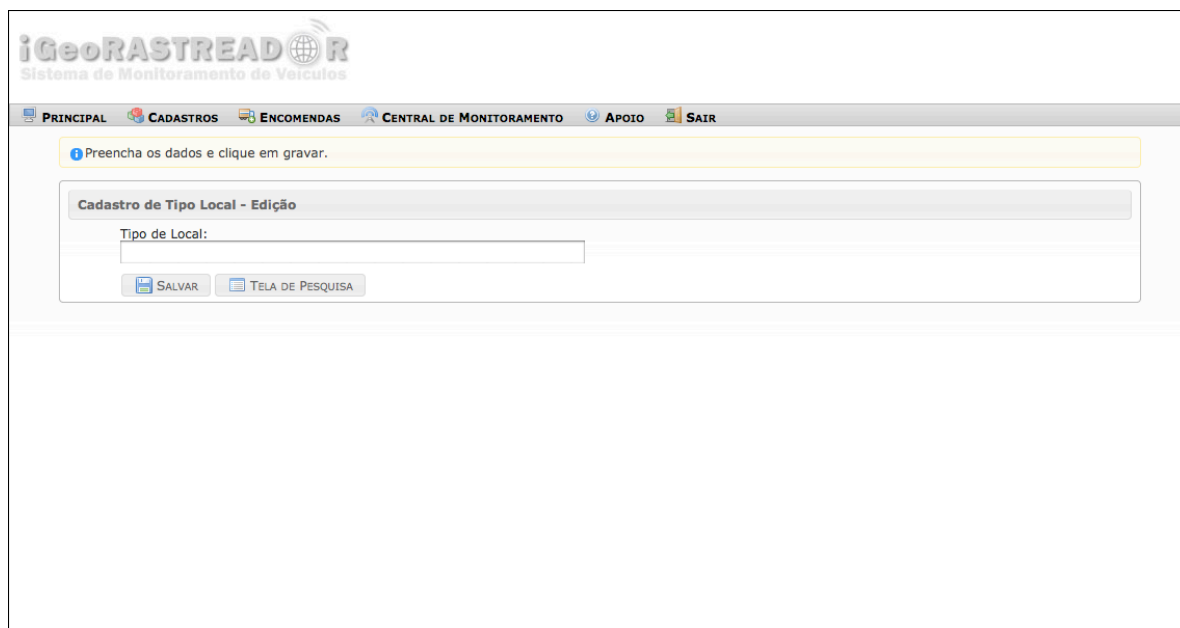


Figura 28 – Formulário de Cadastro dos Tipos de Local.

O cadastro de tipo de local é um simples cadastro auxiliar para complementar informações de especificação ao cadastro de locais. Podem ser cadastrados como tipos: local de entrega, local de partida, ponto de parada e etc.

Figura 29 – Formulário para o Cadastro de Locais

O cadastro de locais é utilizado para cadastro prévio de pontos que serão utilizados para demarcação de percursos. Um mapa interativo é disponibilizado para que seja um facilitador no cadastro do ponto, visto que o que é cadastrado são as coordenadas deste ponto.

Figura 30 – Formulário para Cadastro de Percursos.

O cadastro de percursos faz uso das informações previamente registradas nos cadastros anteriormente mencionados. Um percurso poderá ser cadastrado pelo seu tipo. O tipo *Pontos Exatos*, determina que o percurso cadastrado, é possuidor de um local de início e

um local término. Estes locais são selecionados na caixa de seleção de local início e local fim do formulário. Nestas caixas de seleção, estão listados os locais previamente cadastrados no cadastro de locais. O cadastro de percurso do tipo *Trecho*, é a modalidade do cadastro em que os trechos do percurso são cadastrados por intermédio do mapa interativo.

The screenshot displays the iGeoRASTREADOR web application interface. At the top, the logo 'iGeoRASTREADOR' is visible, followed by the subtitle 'Sistema de Monitoramento de Veículos'. Below this is a navigation bar with links: PRINCIPAL, CADASTROS, ENCOMENDAS, CENTRAL DE MONITORAMENTO, APOIO, and SAIR. A yellow message box states: 'Preencha o filtro e clique em pesquisar ou clique em novo para apresentar a tela de inclusão.' The main content area is titled 'Cadastro de Tipo Ocorrência - Pesquisa'. It features a text input field labeled 'Tipo de Ocorrência:'. Below the input field are two buttons: 'Novo REGISTRO' (with a green plus icon) and 'PESQUISAR' (with a magnifying glass icon). A 'LISTAGEM' button is located below these. At the bottom of the form, a yellow message box indicates: 'Não foram encontrados registros'.

Figura 31 – Formulário de Cadastro para o Tipo de Ocorrência.

O cadastro de tipo de ocorrência é um outro cadastro criado para atender a especificidades do cadastro de Ocorrências. Ele registra os tipos possíveis de ocorrências para os veículos cadastrados no sistema.

The screenshot displays the iGeoRASTREADOR web application interface. At the top, the logo 'iGeoRASTREADOR' is visible with the subtitle 'Sistema de Monitoramento de Veículos'. Below the logo is a navigation bar with tabs: PRINCIPAL, CADASTROS, ENCOMENDAS, CENTRAL DE MONITORAMENTO, APOIO, and SAIR. A yellow message box at the top of the main content area states: 'Preencha o filtro e clique em pesquisar ou clique em novo para apresentar a tela de inclusão.' Below this is a form titled 'Cadastro de Ocorrência - Pesquisa'. The form contains the following fields: 'Veículo:' (a text input field), 'Tipo de Ocorrência:' (a dropdown menu), 'Data da Ocorrência:' (a date picker), and 'Observação:' (a text area). Below these fields is a section labeled 'Lançamento Automático:' containing two buttons: 'NOVO REGISTRO' (with a green plus icon) and 'PESQUISAR' (with a magnifying glass icon). At the bottom of the form is a 'LISTAGEM' button. A yellow status bar at the bottom of the form area displays the message: 'Não foram encontrados registros'.

Figura 32 – Formulário de Cadastro para Ocorrências.

As ocorrências lançadas no sistema são uma parte fundamental do histórico operacional do veículo. Informações sobre acidentes do veículo, trafegar fora do percurso estabelecido, quebras inesperadas e outras, são lançadas neste cadastro. A central de monitoramento também lança ocorrências automaticamente dependendo do seu tipo. Neste protótipo, implementou-se a funcionalidade para lançamento de ocorrência de *Fora do Percurso* caso o veículo esteja efetuando a entrega de uma encomenda pela modalidade de rastreamento por percurso e não esteja cumprindo este percurso previamente cadastrado.

iGeoRASTREADOR
Sistema de Monitoramento de Veículos

PRINCIPAL CADASTROS ENCOMENDAS CENTRAL DE MONITORAMENTO APOIO SAIR

Preencha os dados e clique em gravar.

Cadastro de Encomenda - Edição

Percurso para Entrega:
*
+ -

Cliente:
+ -

Código de Rastreamento:
[]

Descrição da Encomenda:
[]

Observação:
[]

Data de Cadastro:
[]

Entrega Realizada:
[]

Data de Entrega:
[]

[] SALVAR [] TELA DE PESQUISA

Figura 33 – Formulário de Cadastro de Encomendas no Sistema.

O cadastro de encomendas é o segundo cadastro mais importante do sistema. Neste formulário são realizados os cadastros das encomendas. A figura 34 demonstra o diagrama de atividades para este cadastro e em sua rotina também há a descrição do momento em que a central de monitoramento é utilizada para acompanhamento da encomenda.

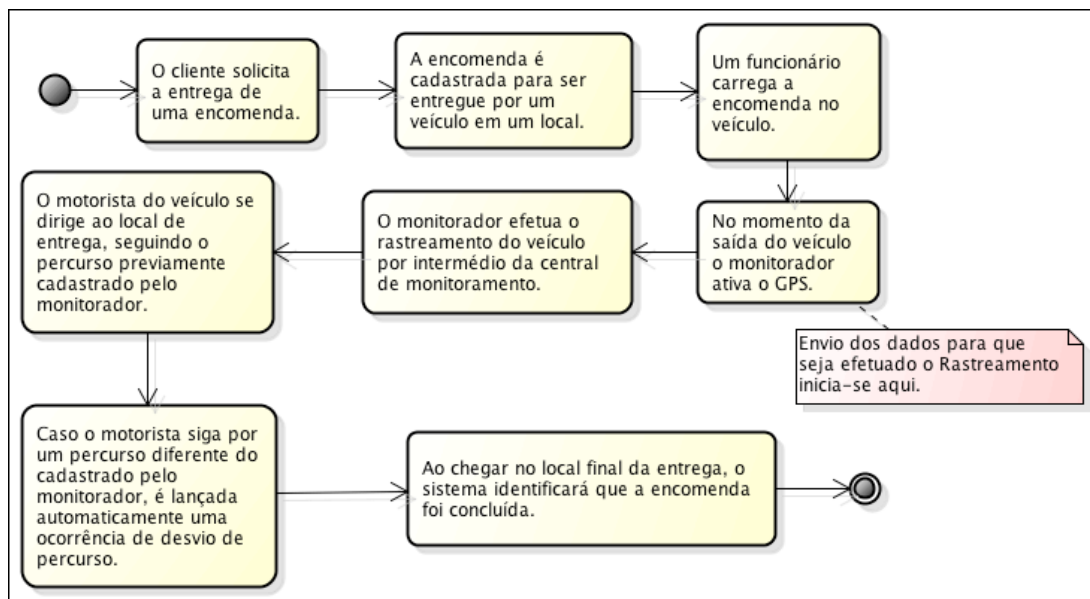


Figura 34 – Diagrama de atividades para o Cadastro de Encomendas

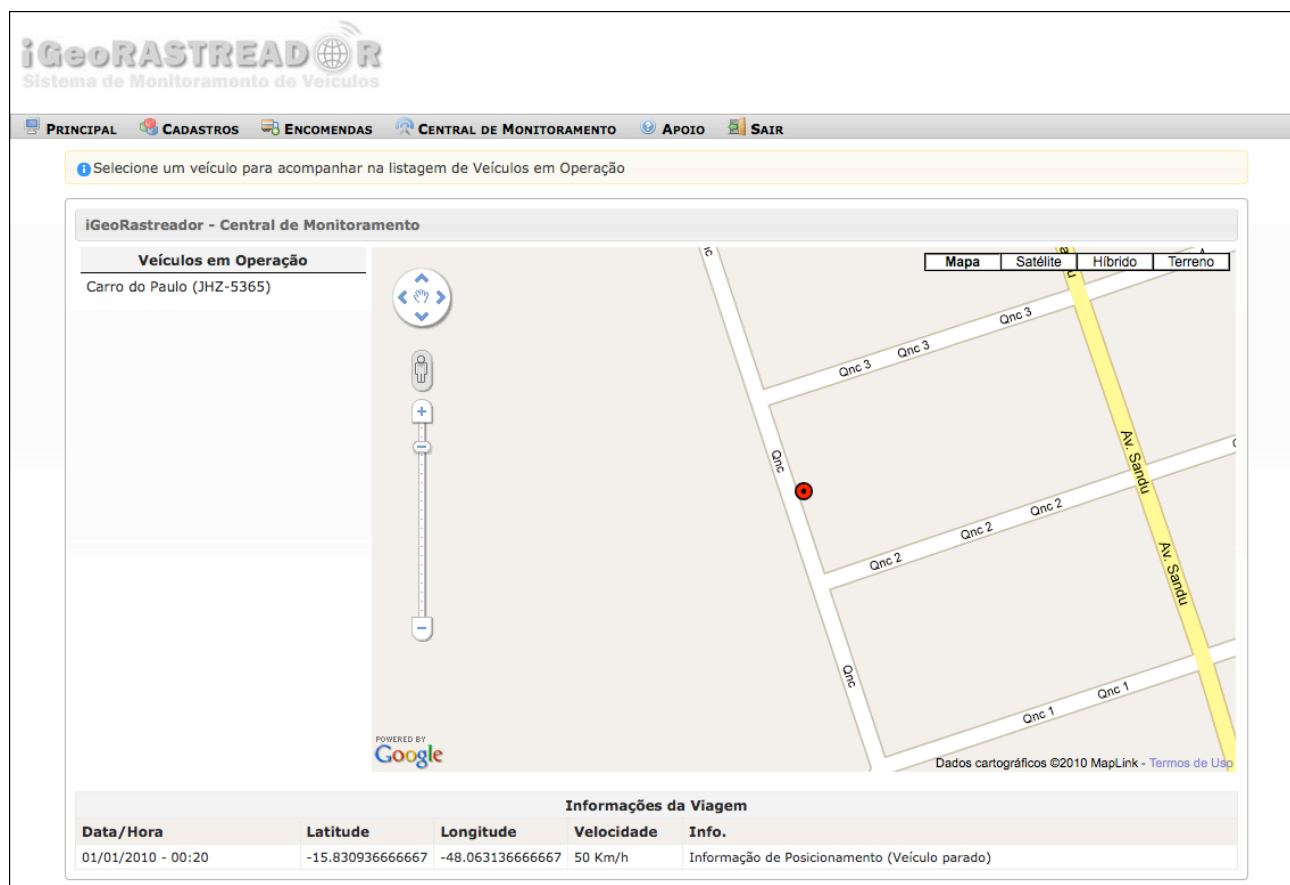


Figura 35 – Central de Monitoramento do Projeto

A figura 35, ilustra a central de monitoramento implementada para o protótipo. Na coluna à esquerda de título *Veículo em Operação*, é possível selecionar algum veículo que esteja em operação no momento do carregamento do programa. A partir daí, será possível verificar informações inerentes a esta viagem na tabela de *Informações da Viagem*. Também será possível a seleção por intermédio do número de uma encomenda cadastrada, para que o veículo que estiver relacionado para o transporte seja exibido no mapa.

O ponto em destaque na cor vermelha da imagem, representa o posicionamento do veículo no mapa, na modalidade de rastreamento livre onde a operação do veículo é acompanhada por um funcionário. Se esta viagem possuísse um percurso relacionado, o percurso estaria traçado no mapa e a localização do veículo neste percurso estaria evidente. Caso este veículo não estivesse cumprindo o percurso, uma ocorrência seria lançada automaticamente e listada na listagem de informações da viagem.

4.3 – DESCRIÇÃO DA IMPLEMENTAÇÃO

Para a implementação do protótipo deste projeto, em primeiro lugar observa-se a necessidade da configuração de um servidor web Apache e o banco de dados PostgreSQL em funcionamento. A tarefa de instalação e configuração de um servidor web não é uma tarefa tão simples, por isto, para testes locais foi adotado o pacote de instalação XAMPP, disponível para todas as plataformas do mercado. O pacote instala e configura automaticamente os programas: Apache, Mysql, PHP, SQLite, phpMyAdmin entre outros. Para o projeto apenas Apache e PHP do pacote foram utilizados. O PostgreSQL é instalado utilizando o pacote de instalação fornecido pela empresa EnterpriseDB. Com isso se tornou possível deixar o sistema funcional no servidor configurado.

Após a instalação dos pacotes, foram configurados os roteadores da residência do autor do projeto de forma que a porta 1234 permanecesse aberta no firewall durante os testes. Isto se fez necessário para receber os pacotes TCP enviados pelo módulo TK-102, a partir da aquisição destes pacotes se tornou possível o desenvolvimento do *importador de dados*. No importador de dados, o problema encontrado diz respeito à conversão das coordenadas obtidas na forma decimal e minutos para o formato aceito pela API de mapas da google. A latitude fornecida pelo módulo TK-102 possui o número no formato: ddmm.mmmm e a longitude dddmm.mmmm, o dd é a parte em degrees (graus) e o m é o número em formato de minutos. Se utilizarmos como exemplo para o calculo da latitude o número 4009.4647, teremos 40 graus e 09.4647 minutos. Para o calculo da longitude a fórmula utilizada é a seguinte:

$$\text{latitude (graus)} = \text{graus} + (\text{minutos} / 60)$$

$$\text{latitude (graus)} = 40 + (09.4674 / 60)$$

$$\text{latitude} = 40 + 0.15779 = \mathbf{40.15779}$$

Para a longitude, utiliza-se fórmula semelhante, porem o grau é formado por 3 caracteres numéricos. Utilizando como exemplo o número 00851.3850, teremos 008 graus e 51.3850 minutos. A fórmula para o calculo da latitude é:

$$\text{longitude (graus)} = \text{graus} + (\text{minutos} / 60)$$

$$\text{longitude (graus)} = 8 + (51.3850 / 60)$$

$$\text{longitude} = 8 + 0.8564166667 = \mathbf{8.8564166667}$$

Caso a latitude esteja no hemisfério Sul (S), o valor deverá ser utilizado com o sinal de negativo. No caso da longitude, o sinal negativo é utilizado caso o hemisfério seja o leste (W).

Um problema encontrado nesta fase do desenvolvimento, foi a correta configuração do módulo para envio das informações conforme estabelecido. O módulo por vezes se demonstra instável quanto ao envio em sequência das informações sob o serviço GPRS. Este módulo escolhido se mostrou eficiente para obtenção de informações de posicionamento do veículo, para que fosse possível o rastreamento. Para agregar valores às informações fornecidas pelo módulo, seria necessária a instalação de sensores no veículo para captura de outras informações por intermédio da telemetria, o que não foi possível com este módulo.

A biblioteca de Google Maps é uma biblioteca de uso simplificado e com uma ampla documentação disponível online. O paradigma mais complexo a ser quebrado dizia respeito a comparação do ponto onde o veículo se encontrava no mapa, com o percurso previamente cadastrado. Para isto, utilizei a fórmula de *haversine*.

A fórmula de *haversine* leva em consideração a circunferência da terra para calcular a distancia entre dois pontos a partir da latitude e a longitude destes dois pontos. Utilizada por navegadores antigos, essa forma é facilitada pelos recursos computacionais atuais. [movable-type.co.uk]

$$A = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(long1) * \cos(lat2) * \sin^2\left(\frac{\Delta long}{2}\right)$$

$$B = 2 * \text{atan2}(\sqrt{A}, \sqrt{1 - A})$$

$$Distância = R * B$$

Figura 36 – Fórmula de haversine

Para a fórmula exibida na figura 36, considera-se R como o maior raio da Terra estabelecido como 6378,137 Km na linha do equador para o DATUM WGS84, padrão do chip GPS do módulo TK-102. DATUM refere-se ao modelo matemático teórico da representação da superfície da Terra ao nível do mar utilizado pelos cartógrafos numa dada carta ou mapa. O Δlat é a diferença entre as duas latitudes utilizadas e o $\Delta long$ é o mesmo padrão, apenas referente à longitude.

Estes cálculos foram necessários para que a central de monitoramento com mapas interativos se tornasse funcional.

CAPÍTULO 5 – EXPERIMENTO E RESULTADOS

5.2 – TESTES REALIZADOS

Os testes foram realizados com informações coletadas em endereços próximos à residência do autor do projeto. Após a instalação do aparelho GPS no veículo, o autor realizou viagens com o intuito de colher informações e testar o funcionamento da central de monitoramento.

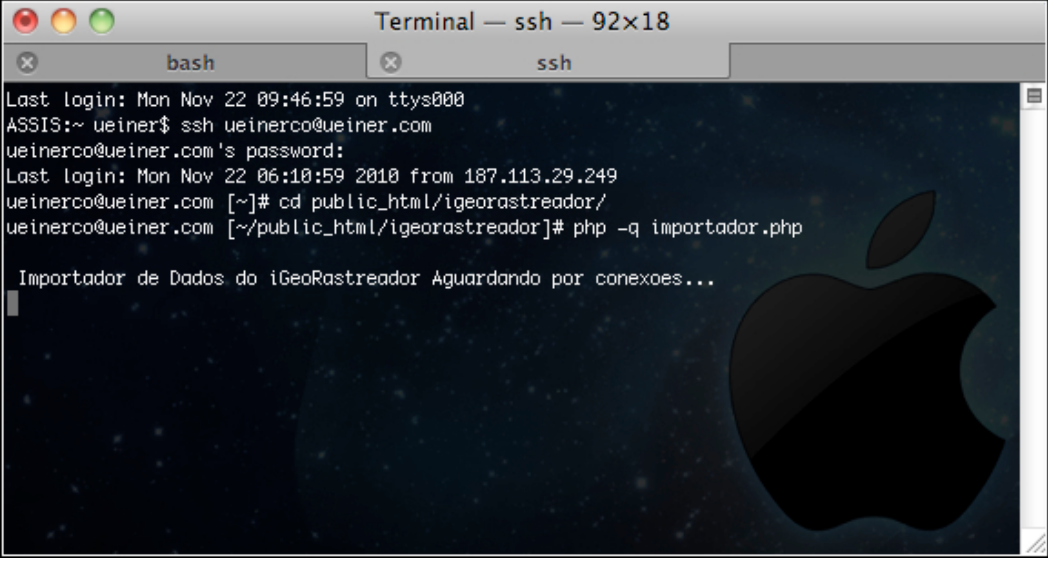
Foram necessárias três etapas para a conclusão dos testes, sendo a primeira etapa, a etapa de coleta de dados e inserção no banco de dados com a utilização do módulo TK-102 juntamente com o script *importador de dados* implementado. A segunda etapa diz respeito a utilização da central de monitoramento para um rastreamento “livre”, onde houve o acompanhamento do veículo por intermédio da central de monitoramento. Para conclusão dos testes, houve a necessidade do cadastro de um percurso, e o veículo percorreu este trecho definido. Durante este teste, caso o veículo não estivesse percorrendo este trecho, uma ocorrência deveria ser lançada com o intuito de informar o não cumprimento do percurso por parte do motorista.

Para que os testes finais relacionados à central de monitoramento pudessem ser realizados, os cadastros auxiliares foram utilizados para registrar informações que seriam necessárias. Dados como clientes, veículos, percursos, ocorrências foram cadastrados inicialmente no sistema.

5.2.1 – ROTINA PARA TESTES DO IMPORTADOR DE DADOS

Este teste tem como objetivo demonstrar que as informações provenientes do módulo TK-102 são capturadas por intermédio do importador de dados e inseridas na tabela *gps_tracker* do banco de dados do sistema.

Após a inicialização do via linha de comando, o importador exibe a mensagem de que está sendo executado e aguardando conexões. Neste momento, qualquer mensagem enviada pelo GPS é capturada, manipulada e inserida no banco de dados.

A screenshot of a macOS Terminal window titled "Terminal — ssh — 92x18". The window has two tabs: "bash" and "ssh". The terminal output shows a successful SSH login to a server named "ueinerco@ueiner.com". The user enters their password, and the terminal displays the last login time and IP address. The user then navigates to the directory "/public_html/igeorastreador/" and runs the command "php -q importador.php". The output of the script is "Importador de Dados do iGeoRastreador Aguardando por conexoes...". The terminal background features a dark space theme with a large, faint Apple logo on the right side.

```
Terminal — ssh — 92x18
bash ssh
Last login: Mon Nov 22 09:46:59 on ttys000
ASSIS:~ ueiner$ ssh ueinerco@ueiner.com
ueinerco@ueiner.com's password:
Last login: Mon Nov 22 06:10:59 2010 from 187.113.29.249
ueinerco@ueiner.com [~]# cd public_html/igeorastreador/
ueinerco@ueiner.com [~/public_html/igeorastreador]# php -q importador.php

Importador de Dados do iGeoRastreador Aguardando por conexoes...
```

Figura 37 – Terminal executando o importador de dados do sistema

Nos testes realizados, foram obtidos registros de pontos da quadra QNC, em Taguatinga Norte/DF. Estes pontos foram utilizados posteriormente para plotagem do mapa da central de monitoramento.

A figura 38, demonstra o relatório da importação que é emitido para cada mensagem recebida e inserida no banco de dados.

```

Terminal — ssh — 132x45
ssh
ueinerco@ueiner.com [~/public_html/igeorastreador]# php -q importador.php

Importador de Dados do iGeoRastreador Aguardando por conexoes...

iGeoRastreador: Conexao com o GPS Iniciada!

Mensagem Recebida: 101122120200,+556184042241,GPRMC,120200.000,A,1549.8593,S,04803.7906,W,0.00,0.00,221110,,,A*68,F,, imei:01141200
1070485,08,1165.7,F:3.92V,0,140,30603,,,01CD,0855

iGeoRastreador: Dados Inseridos no Banco com Sucesso!

Fim da Mensagem! - iGeoRastreador

Mensagem Recebida: 101122120218,+556184042241,GPRMC,120218.000,A,1549.8596,S,04803.7908,W,0.00,0.00,221110,,,A*6A,F,, imei:01141200
1070485,08,1165.7,F:3.89V,0,140,28733,,,01CD,0855

iGeoRastreador: Dados Inseridos no Banco com Sucesso!

Fim da Mensagem! - iGeoRastreador

Mensagem Recebida: 101122120250,+556184042241,GPRMC,120250.000,A,1549.8597,S,04803.7908,W,0.00,0.00,221110,,,A*67,F,, imei:01141200
1070485,08,1162.3,F:3.90V,0,140,12006,,,01CD,0855

iGeoRastreador: Dados Inseridos no Banco com Sucesso!

Fim da Mensagem! - iGeoRastreador

Mensagem Recebida: 101122120320,+556184042241,GPRMC,120320.000,A,1549.8632,S,04803.7873,W,0.00,144.66,221110,,,A*61,F,, imei:011412
001070485,08,1169.7,F:3.89V,0,142,34943,,,01CD,0855

iGeoRastreador: Dados Inseridos no Banco com Sucesso!

Fim da Mensagem! - iGeoRastreador

Mensagem Recebida: 101122120350,+556184042241,GPRMC,120350.000,A,1549.8742,S,04803.7818,W,2.11,111.60,221110,,,A*69,F,, imei:011412
001070485,08,1170.6,F:3.90V,0,142,43992,,,01CD,0855

iGeoRastreador: Dados Inseridos no Banco com Sucesso!

```

Figura 38 – Dados inseridos no banco de dados a partir de mensagens recebidas do módulo TK-102

Abaixo, a figura 39 ilustra os 23 primeiros registros inseridos na tabela *gps_tracker* do banco de dados após o início do rastreamento.

	gps_id_	gps_nr_	gps_cs_t	gps_nr_tempo_u	gp	gps_nr_latitu	gp	gps_cs_longiti	gp	gps_nr	gps_cs_cu	gps_dt_gps	gps_nr	gps_nr_imei
	[PK] ser	character varying(25)	character	character varying(25)	ch	character var	ch	character vary	ch	character	character	character v	character	character varying(25)
1	3	+556184042241	GPRMC	120200.000	A	1549.8593	S	04803.7906	W	0.00	0.00	221110	A*68	011412001070485
2	4	+556184042241	GPRMC	120218.000	A	1549.8596	S	04803.7908	W	0.00	0.00	221110	A*6A	011412001070485
3	5	+556184042241	GPRMC	120250.000	A	1549.8597	S	04803.7908	W	0.00	0.00	221110	A*67	011412001070485
4	6	+556184042241	GPRMC	120320.000	A	1549.8632	S	04803.7873	W	0.00	144.66	221110	A*61	011412001070485
5	7	+556184042241	GPRMC	120350.000	A	1549.8742	S	04803.7818	W	2.11	111.60	221110	A*69	011412001070485
6	8	+556184042241	GPRMC	120421.000	A	1549.7878	S	04803.8159	W	15.85	341.80	221110	A*50	011412001070485
7	9	+556184042241	GPRMC	122027.000	A	1549.8503	S	04803.7876	W	0.54	49.64	221110	A*5C	011412001070485
8	10	+556184042241	GPRMC	122051.000	A	1549.8534	S	04803.7876	W	0.82	178.95	221110	A*6F	011412001070485
9	11	+556184042241	GPRMC	122121.000	A	1549.8525	S	04803.7880	W	0.00	178.95	221110	A*6A	011412001070485
10	12	+556184042241	GPRMC	122206.000	A	1549.8476	S	04803.7871	W	0.78	165.88	221110	A*6A	011412001070485
11	13	+556184042241	GPRMC	122231.000	A	1549.8492	S	04803.7887	W	1.05	185.96	221110	A*67	011412001070485
12	14	+556184042241	GPRMC	122302.000	A	1549.8359	S	04803.7878	W	4.52	218.00	221110	A*69	011412001070485
13	15	+556184042241	GPRMC	122333.000	A	1549.8618	S	04803.7868	W	0.00	173.95	221110	A*6B	011412001070485
14	16	+556184042241	GPRMC	122403.000	A	1549.8623	S	04803.7866	W	0.00	173.95	221110	A*69	011412001070485
15	17	+556184042241	GPRMC	122621.000	A	1549.8638	S	04803.7884	W	0.00	173.95	221110	A*6D	011412001070485
16	18	+556184042241	GPRMC	122646.000	A	1549.8637	S	04803.7885	W	0.00	173.95	221110	A*62	011412001070485
17	19	+556184042241	GPRMC	122711.000	A	1549.8642	S	04803.7890	W	0.00	173.95	221110	A*67	011412001070485
18	20	+556184042241	GPRMC	122742.000	A	1549.8637	S	04803.7884	W	0.00	173.95	221110	A*66	011412001070485
19	21	+556184042241	GPRMC	122812.000	A	1549.8630	S	04803.7878	W	0.00	173.95	221110	A*68	011412001070485
20	22	+556184042241	GPRMC	122842.000	A	1549.8630	S	04803.7878	W	0.00	173.95	221110	A*6D	011412001070485
21	23	06198258065	GPRMC	122906.000	A	1549.8631	S	04803.7879	W	0.00	173.95	221110	A*6C	011412001070485
22	24	06198258065	GPRMC	122924.000	A	1549.8626	S	04803.7873	W	0.00	173.95	221110	A*60	011412001070485
23	25	06198258065	GPRMC	122948.000	A	1549.8622	S	04803.7870	W	0.00	173.95	221110	A*6D	011412001070485

Figura 39 – Informações inseridas no banco, obtidas por intermédio do módulo TK-102.

5.2.2 – ROTINA PARA TESTES DA CENTRAL DE MONITORAMENTO SEM PERCURSO DEFINIDO

Para realização deste teste, foram utilizados pontos obtidos na quadra QNC, em Taguatinga Norte/DF. O aparelho GPS estava acomodado no veículo e ativo. Ao acessar a central de monitoramento, o veículo teve sua localização centralizada no mapa.

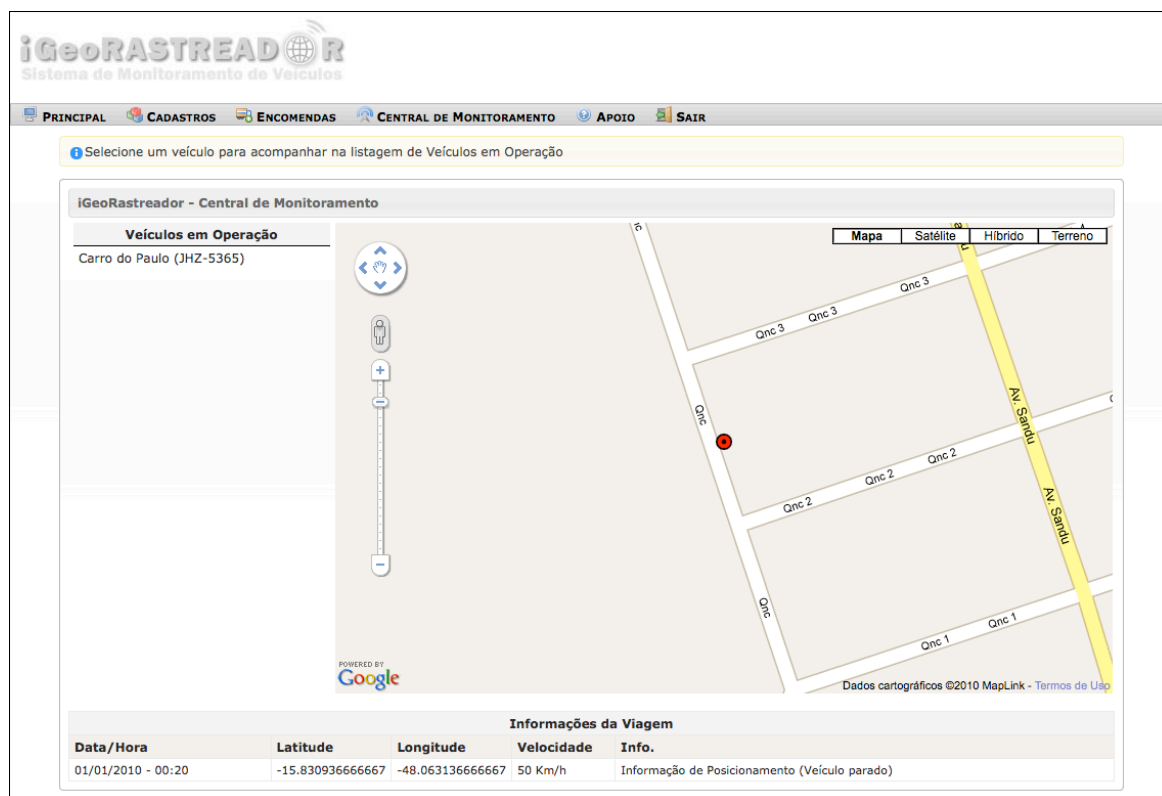


Figura 40 – Veículo sendo rastreado por intermédio da central de monitoramento

Após o início de sua locomoção, os pontos do percurso por ele percorrido foram marcados no mapa, sendo o ponto em vermelho, o último ponto em que o veículo se encontra e o amarelo os pontos por onde o veículo passou. Esta situação pode ser observada na figura 41.

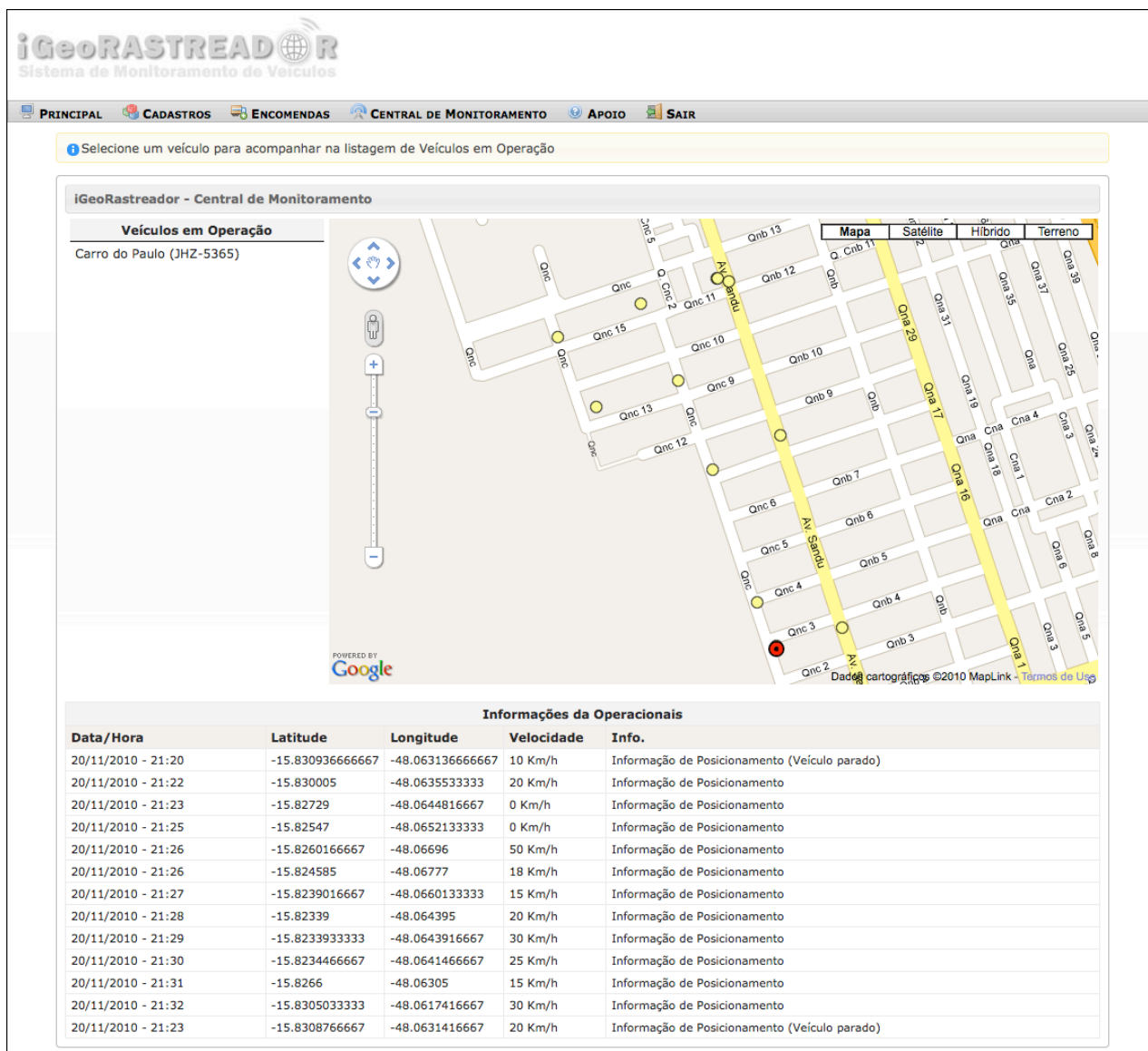


Figura 41 – Veículo monitorado por intermédio da Central de Monitoramento

A central de monitoramento se comportou como esperado, na situação do teste de monitoramento sem um percurso definido, observa-se na imagem um percentual de erro elevado para exibição dos pontos no mapa, devido à baixa precisão do GPS integrado ao módulo TK-102. Um outro fator a ser considerado para a ocorrência deste erro é a precisão utilizada para arredondamento do número das coordenadas obtidas por intermédio do módulo.

5.2.3 – ROTINA PARA TESTES DA CENTRAL DE MONITORAMENTO COM PERCURSO DEFINIDO

Para este teste, previamente cadastrou-se um percurso por intermédio do cadastro de percursos do sistema. Nesta modalidade de rastreamento, se faz necessário que uma encomenda seja cadastrada e o percurso cadastrado anteriormente selecionado. Ao acessar a central de monitoramento, haverá a informação da encomenda em rastreamento.

O percurso utilizado possui seu ponto inicial na QNC 2 e o final QNG 22, ambos os pontos em Taguatinga Norte/DF. A figura 42 ilustra o resultado do teste quanto a apresentação do percurso percorrido no mapa.

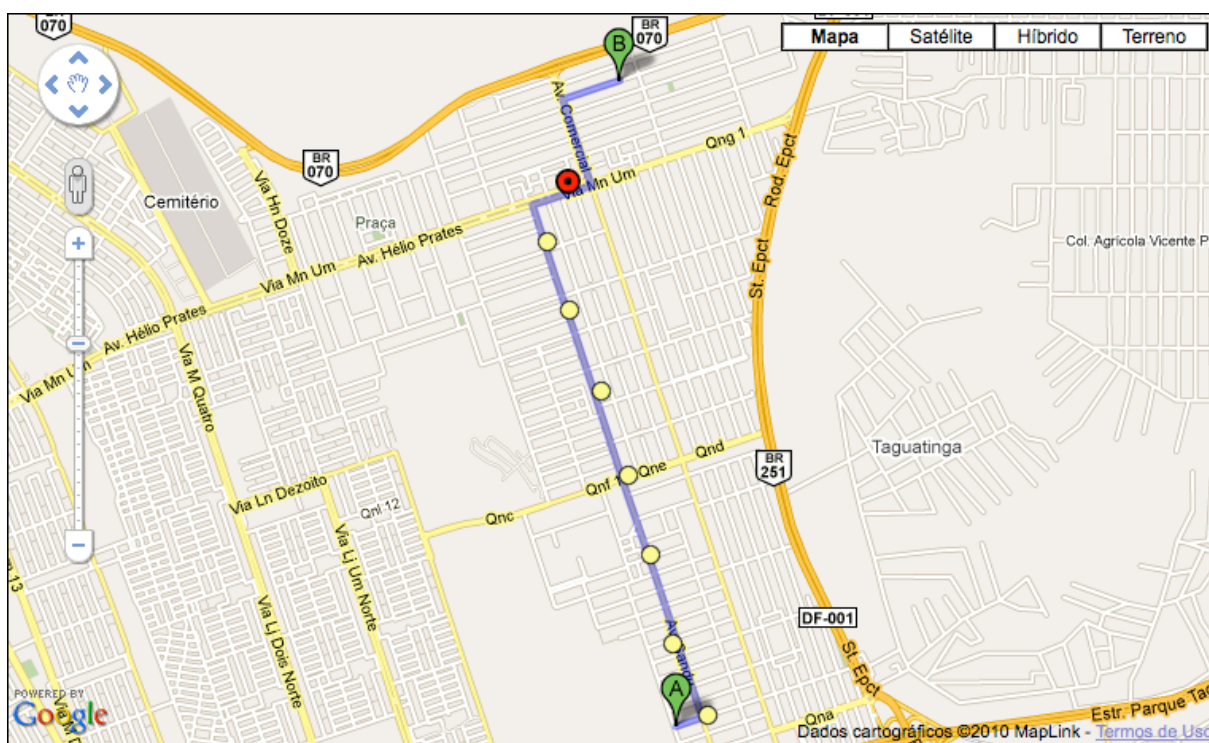


Figura 42 – Modelo de rastreamento por um percurso previamente cadastrado.

O raio utilizado para comparação da passagem do veículo no percurso cadastrado é de 10 metros. Uma ocorrência de desvio de percurso é lançada a cada vez que o veículo não cumprir o percurso proposto. Este recurso não se mostrou completamente eficiente devido à imprecisão do módulo GPS utilizado.

5.3 – RESULTADOS DA APLICAÇÃO DO MODELO

Os cenários de testes adotados se mostraram eficientes para a demonstração das funcionalidades do protótipo para sistema de rastreamento desenvolvido neste projeto.

O importador de dados é um script flexível que pode ser adaptado para recepção de mensagens de qualquer aparelho GPS que utilize o serviço GPRS para envio de pacotes TCP, basta que o módulo PHP esteja ativo no servidor Apache e a funcionalidade de *sockets* ativada.

Para o correto funcionamento da central de monitoramento, apenas é necessário que a tabela *gps_tracker* possua informações de coordenadas geográficas do veículo que se deseja rastrear. Uma limitação do sistema é a relacionada a informações sobre o estado do veículo, visto que o módulo GPS adotado não é passível de integração a sensores que poderiam agregar maior valor às informações obtidas.

A central de monitoramento é o cérebro do sistema, e nos testes se mostrou eficiente para exibição de informações sobre o posicionamento do veículo. A configuração efetuada no GPS para o envio de mensagens para o servidor a cada 30 se mostrou suficiente para o correto funcionamento do sistema.

CAPÍTULO 6 - CONCLUSÃO

6.1 – CONCLUSÕES

Este projeto apresenta o desenvolvimento de um sistema que fornece informações sobre posicionamento dos veículos da frota de uma empresa de transporte por intermédio da navegação por satélite. Sistemas que efetuam tal tarefa são conhecidos como sistema de rastreamento de veículos. O sistema proposto faz uso de um módulo GPS integrado a um modem GSM que sob o serviço GPRS envia as mensagens num intervalo de tempo pré-determinado de tempos em tempos a um servidor que possui um serviço ativo para recebê-las, efetuar o tratamento e a posterior inserção das mesmas no banco de dados do sistema.

No servidor, o sistema iGeoRastreador está instalado e será a infraestrutura que viabilizará a exploração dos acontecimentos operacionais da empresa no dia-a-dia. O sistema é dotado de cadastros auxiliares que são mantidos por um usuário administrador. Estes cadastros auxiliam o controle das informações por parte dos funcionários e também tem suas informações utilizadas pela central de monitoramento, que é a responsável por fornecer em forma de mapas, dados sobre o posicionamento do veículo.

O projeto obteve êxito ao implementar um protótipo do sistema que fornece informações sobre o posicionamento dos veículos, um sistema de rastreamento de veículos. Todos os objetivos propostos e especificados foram atingidos, com a ressalva de que o módulo GPS adotado não teve seu funcionamento como o esperado. O mesmo não manteve um padrão de intervalo de tempo para que as mensagens fossem enviadas ao servidor especificado. Para um sistema corporativo, o mesmo não seria viável pois não existiria uma confiabilidade nas informações obtidas, podendo ocorrer erros durante a operação do sistema. Sugere-se a aquisição de um módulo GPS que seja específico para este tipo de atividade e que aceite a integração a sensores que serão instalados no veículo.

Constantemente vários sistemas com as mesmas características propostas neste projeto, surgem no mercado e auxiliam empresas que possuem frotas de veículos em sua gestão. Caso estes sistemas venham a ser integrados aos sistemas de gestão internos da empresa, independente das tecnologias adotadas, podemos afirmar que haverá redução de custos com mão de obra, aumento da segurança no transporte de mercadorias, e uma melhoria no planejamento logístico da empresa podendo até mesmo obter um aumento nos seus lucros. É neste cenário que poderemos inserir um sistema do gênero do proposto neste projeto.

6.2 – SUGESTÕES PARA TRABALHOS FUTUROS

Como sugestão para trabalhos subsequentes sugere-se a melhoria no importador de dados para não haja a necessidade do mesmo ser executado no servidor por intermédio de um terminal shell, sob o protocolo SSH.

Outra sugestão seria utilizar um módulo GPS com maior precisão e que agregue outras funcionalidades, visto que o módulo utilizado não se mostrou tão eficiente para o sistema do projeto proposto. Com um módulo de maior precisão, a funcionalidade de lançamento de ocorrências para os veículos poderia ser aplicada com maior eficiência. Este novo módulo, deverá ser passível de integração a sensores instalados no veículos para assim permitir que o sistema de rastreamento não apenas forneça informações sobre seu posicionamento mas também informações sobre o estado do veículo, como: bloqueio da ignição, calibragem dos pneus, consumo de combustível, portas abertas, limitação de velocidade por intermédio da telemetria e etc (outros sensores poderão ser utilizados dependendo do ramo de atividade em que a empresa atua e o tipo de veículo que utiliza).

Também seria uma sugestão, a criação de um cadastro de rotas onde haverá um script que irá efetuar a sugestão da melhor rota a ser seguida pelo veículo, baseada em fatores a serem definidos numa etapa de levantamento de requisitos. Este roteamento deverá se efetuado antes mesmo da ocorrência da viagem.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, Sérgio. **A Matemática do GPS**. Revista do Professor de Matemática - n.59, p.17-26, jan. / abr. 2006.
- APPLEGATE, David L. [et al.] – **The travelling salesman problem: a computational Study**. Princeton University Press, 2006.
- CUGNASCA, C. E.; PAZ, S. M. **O Sistema de Posicionamento Global (GPS) e suas Aplicações**. Boletim Técnico da Escola Politécnica da USP, São Paulo, n.BT/PCS/970, 1997.
- CUNHA, Claudio Barbieri da; BONASSER, Ulisses de Oliveira; ABRAHÃO, Fernando Teixeira Mendes – **Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante [Em Linha]**. ANPET, 2002. [Acesso em 09 Out. 2010]. Disponível em WWW: <URL: http://www.ptr.poli.usp.br/ptr/docentes/cbcunha/files/2-opt_TSP_Anpet_2002_CBC.pdf>
- EL-RABBANY, Ahmed. **Introduction to GPS, The Global Positioning System**. Artech House Inc, 2002.
- FABRI, Sergio Mauri; FERREIRA, Luiz Danilo Damasceno. **Sistema Global de Posicionamento (GPS): Introdução e Aplicações Básicas**. TUIUTI: Ciência e Cultura - v.14 n.14, p.115-124, jan. / jan. 2000.
- FIGUEIREDO, Luiz Carlos. **Modelagem Matemática de Sistemas de Rastreamento de Veículos**. DOXA: Revista Semestral do UNILESTE - v.3, n.7, p.17-2, jan. / jun. 2002.
- GOODMAN, Danny. **JAVASCRIPT: A BIBLIA**, RIO DE JANEIRO: CAMPUS, 2001.
- GUTMANS, Andi Gutmans; BAKKEN, Stig; RETHANS, Derick. **PHP 5 – Power Programming**. PRENTICE HALL, 2004.
- KABIR, Mohammed J. **APACHE SERVER 2: A BIBLIA**, RIO DE JANEIRO: CAMPUS , 2002.
- MCNAMARA, Joel. **GPS for Dummies, 2nd Edition**. Wiley Publishing Inc, 2004.
- MILANI, André. **PostgreSQL – Guia do Programador**. NOVATEC, 2008.
- MINETTO, Elton Luís. **Frameworks para Desenvolvimento em PHP**. Novatec, 2007.
- NIEDERAUER, Juliano. **Desenvolvimento Websites com PHP: Aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados**. São Paulo: Novatec, 2004.
- PURVIS, Michael; SAMBELLS, Jeffrey; TURNER Cameron. **Beginning Google Maps Applications with PHP and Ajax: From Novice to Professional**. Apress, 2006.

RODRIGUES, Marcos; CUGNASCA, Carlos Eduardo; FILHO, Alfredo Pereira de Queiroz.

Rastreamento de Veículos. Oficina de Textos, 2009.

UDELL, Sterling. **Beginning Google Maps Mashups with Mapplets, KML, and GeoRSS:**

From Novice to Professional. Apress, 2008.

WINGE, M. et. al. 2001. **Glossário Geológico Ilustrado.** Publicado na Internet:

<http://www.unb.br/ig/glossario/> e disponível em 16 de outubro de 2010.

[SITES]

ACHOUR, Mehdi et AL. **“Manual PHP”** http://www.php.net/manual/pt_BR/index.php

Acesso em: 07/09/2010.

APACHE FOUNDATION **“Apache HTTP Project”** <http://httpd.apache.org/>

Acesso em: 08/09/2010.

MUNDO GEO **“Geoinformação para todos”** <http://www.mundogeo.com.br/>

Acesso em: 08/09/2010.

FRAMECALIXTO **“Framework em PHP desenvolvido para agilizar e simplificar a construção de aplicativos WEB”** <http://code.google.com/p/framecalixto/>

Acesso em: 08/09/2010.

MINISTÉRIO DOS TRANSPORTES **“Portal do Governo”** <http://www.transportes.gov.br/>

Acesso em: 09/09/2010.

GOOGLEMAPS **“API de mapas da Google”**

<http://code.google.com/intl/pt-BR/apis/maps/index.html>

Acesso em: 09/09/2010.

XEXUN **“Fabricante GPS Tracker TK-102”**

<http://www.xexun.com/eshowcps.asp?id=57&style=GPS%B6%A8%CE%BB%D7%B7%D7%D9%C6%F7>

Acesso em: 09/09/2010.

GUIA DO TRANSPORTADOR **“Portal”** <http://www.guiadotrc.com.br/>

Acesso em 09/09/2010.

REDES GSM, GPRS, EDGE E UMTS **“Ricardo Di Lucia Santos”**

http://www.gta.uff.br/ensino/eel879/trabalhos_vf_2008_2/ricardo/1_2.html

Acesso em 27/09/2010.

CONSEGURANÇA **“Soluções em Rastreamento de Frotas”**

<http://rastreamentodefrotas.conseguranca.com.br/>

Acesso em 27/09/2010.

PAULO DRUGOS “CSS” <http://drugos.com.br/Conteudo-Web/css.html>

Acesso em 29/09/2010

DHTML “Dynamic HTML” <http://dicionariodainternet.com.br/cgi-bin/wiki.pl?DHTML>

Acesso em 29/09/2010

JAVASCRIPT LIBRARY “jQuery” <http://jquery.com/>

Acesso em 29/09/2010

W3C MARKUP “Linguagens de Marcação” <http://www.w3.org/MarkUp/>

Acesso em 29/09/2010

GOOGLE MAPS API “Javascript API”

<http://code.google.com/intl/pt-BR/apis/maps/documentation/javascript/>

Acesso em 09/10/2010

OPTIMAP “Behind the Scenes of OptiMap” <http://gebweb.net/blogpost/2007/07/05/behind-the-scenes-of-optimap/>

Acesso em 09/10/2010

DOCUMENTAÇÃO KML “REFERÊNCIA DO KML”

<http://code.google.com/intl/pt-BR/apis/kml/documentation/kmlreference.html>

Acesso em 05/10/2010

WIRELESS BR “Introdução ao GPS”

http://www.wirelessbrasil.org/wirelessbr/colaboradores/thienne_johnson/gps.html

Acesso em 16/10/2010

CERNE “Tutorial GPS” <http://www.cerne-tec.com.br/TutorialGPS.pdf>

Acesso em 16/10/2010

OVERCAR “Como Funciona o GPS” <http://www.overcar.com.br/como-funciona-o-gps/>

Acesso em 16/10/2010

JRANK.ORG “How do i use globes and maps”

<http://www.jrank.org/history/pages/8345/How-Do-I-Use-Globes-Maps.html>

Acesso em 16/10/2010

NMEA “NMEA 0183” <http://www.kh-gps.de/nmea-faq.htm>

Acesso em 16/10/2010

GALILEO INFORMATION CENTER “FOR LATIN AMERICAN”

<http://www.galileoic.org/la/?q=node/190#1>

Acesso em 16/10/2010

APÊNDICE A – CÓDIGO DO IMPORTADOR DE DADOS

```
#!/usr/bin/php -q
<?php

// Tornar o tempo de execucao ilimitado
set_time_limit (0);

//include( ".sistema/debug.php" );
//$stEntrada
=
"101017193522,06184042241,GPRMC,193522.000,A,1549.8564,S,04803.7881,W,0.00,36.7
6,171010,,,A*52,F,,imei:011412001070485,05,1209.7,F:3.92V,0,139,16845,,,01CD,0855";

// Setando variaveis de IP e Porta onde o listener recebera as informacoes do GPS
$host = "69.195.78.117";
$porta = 1234;

// Criando o Socket
$socket = socket_create(AF_INET, SOCK_STREAM, 0) or die("\n NAO FOI POSSIVEL
CRIAR O SOCKET!! - iGeoRastreador \n");

// Vincular o socket criado a porta
$rsSocket = socket_bind($socket, $host, $porta) or die("\n Nao Foi Possivel Vincular a Porta
ao Socket Criado!! - iGeoRastreador \n");

// Iniciar a "escuta" por conexoes
$rsSocket = socket_listen($socket, 3) or die("\n Impossivel iniciar a escuta no Socket!! -
iGeoRastreador \n");

echo "\n Importador de Dados do iGeoRastreador Aguardando por conexoes... \n";

// aceitar conexoes de entrada
```



```
$socketGPS = socket_accept($socket) or die("\n Nao Foi Possivel Aceitar uma Conexão de
Entrada!! - iGeoRastreador \n");
```

```
// Escreve uma mensagem de afirmação quanto ao pedido de ligação recebido
echo "\n iGeoRastreador: Conexao com o GPS Iniciada! \n";
```

```
// Escreve uma mensagem de boas vindas ao cliente
$stBemVindo = "\n Integrador do iGeoRastreador Iniciado! \n";
socket_write($socketGPS, $stBemVindo, strlen ($stBemVindo)) or die("\n Nao Foi Possivel
Enviar a Mensagem!! - iGeoRastreador \n");
```

```
// Manter o loop verificando entrada de informacoes por parte do cliente
do
{
```

```
    // Lendo a entrada do cliente
    $stEntrada = socket_read($socketGPS, 1024, 1) or die("Impossivel Ler a Mensagem
Enviada Pelo Cliente!! - iGeoRastreador \n");
    $stringGps = explode( ' ', $stEntrada );
```

```
    //Verifica se a string nao esta vazia e se os a primeira parte da string possui 12 caracteres
    if ( trim($stEntrada) != "" && ( count($stringGps) && strlen($stringGps[0]) == 12 ) )
    {
```

```
        //Prepara a string de dados para ser inserida no banco.
```

```
        echo "\n Mensagem Recebida: $stEntrada \n";
```

```
$arDados = array();
```

```
$arDados['nr_serie']          = $stringGps[0];
$arDados['telefone_authorized'] = $stringGps[1];
$arDados['tipo_mensagem']     = $stringGps[2];
$arDados['tempo_utc']         = $stringGps[3];
```

```

$arResultados['estado']          = $stringGps[4];
$arResultados['latitude']        = $stringGps[5];
$arResultados['latitude_tipo']    = $stringGps[6];
$arResultados['longitude']       = $stringGps[7];
$arResultados['longitude_tipo']   = $stringGps[8];
$arResultados['velocidade_nos']   = $stringGps[9];
$arResultados['curso']           = $stringGps[10];
$arResultados['data_ddmmyy']      = $stringGps[11];
$arResultados['variacaomagnetica'] = $stringGps[12];
$arResultados['modo_gps']         = $stringGps[13];
$arResultados['checksum']         = $stringGps[14];
$arResultados['indicador_sinal_gps'] = $stringGps[15];
$arResultados['imei']             = str_replace( " imei:", "", $stringGps[17] );
$arResultados['numero_gps']       = $stringGps[18];
$arResultados['altitude']         = $stringGps[19];
$arResultados['forca_bateria']     = $stringGps[20];
$arResultados['carregando']       = $stringGps[21];
$arResultados['tamanho_string']   = $stringGps[22];
$arResultados['crc_checksum']     = $stringGps[23];
$arResultados['codigo_pais']      = $stringGps[24];
$arResultados['codigo_rede']      = $stringGps[25];
$arResultados['lac']              = $stringGps[26];
$arResultados['cell_id']          = $stringGps[27];

//x($stringGps,$arResultados);die();

//inicia a conexão com o banco de dados

$string_conexao = "host=localhost dbname=ueinerco_projeto final user=ueinerco_projeto
password=projeto";

$conexao = pg_connect($string_conexao);

$sqlInserir = "

```

```
INSERT INTO gps_tracker
(
    gps_nr_serial,
    gps_nr_authorized,
    gps_cs_tipo_mensagem,
    gps_nr_tempo_utc,
    gps_cs_estado_sinal,
    gps_nr_latitude,
    gps_cs_latitude_tipo,
    gps_cs_longitude,
    gps_cs_longitude_tipo,
    gps_nr_velocidade,
    gps_cs_curso,
    gps_dt_gps,
    gps_vl_variacao_magnetica,
    gps_cs_modulo_gps,
    gps_nr_checksum_gprmc,
    gps_cs_indicador_gps,
    gps_nr_imei,
    gps_nr_gps,
    gps_nr_altitude,
    gps_vl_bateria,
    gps_cs_carregando,
    gps_nr_tamanho_string,
    gps_nr_checksum,
    gps_nr_codigo_pais,
    gps_nr_codigo_rede,
    gps_nr_codigo_area,
    gps_nr_id_celular
)
VALUES
(
    '${arDados['nr_serie']}',
    '${arDados['telefone_authorized']}',
```

```

'{$arDados['tipo_mensagem']}',
'{$arDados['tempo_utc']}',
'{$arDados['estado']}',
'{$arDados['latitude']}',
'{$arDados['latitude_tipo']}',
'{$arDados['longitude']}',
'{$arDados['longitude_tipo']}',
'{$arDados['velocidade_nos']}',
'{$arDados['curso']}',
'{$arDados['data_ddmmyy']}',
'{$arDados['variacaomagnetica']}',
'{$arDados['modo_gps']}',
'{$arDados['checksum']}',
'{$arDados['indicador_sinal_gps']}',
'{$arDados['imei']}',
'{$arDados['numero_gps']}',
'{$arDados['altitude']}',
'{$arDados['forca_bateria']}',
'{$arDados['carregando']}',
'{$arDados['tamanho_string']}',
'{$arDados['crc_checksum']}',
'{$arDados['codigo_pais']}',
'{$arDados['codigo_rede']}',
'{$arDados['lac']}',
'{$arDados['cell_id']}'
)

";

//xd($sqlInserir);die();

$inserir = pg_query( $sqlInserir );

if( $inserir ){

```

```

        // Escreve uma mensagem de afirmação quanto a insercao no banco de dados
        echo "\n iGeoRastreador: Dados Inseridos no Banco com Sucesso! \n";
    }
    // Para dar fim a sessao, é necessário que o cliente envie a mensagem FIM
    if (trim($stEntrada) == "FIM")
    {

        // Fechar o socket "filho"
        socket_close($socketGPS);
        // parar o loop
        break;

    } else {

        // Fecha o socket primario
        socket_close($socketGPS);
        echo "\n Fim da Mensagem! - iGeoRastreador \n";

        // Permite que o socket aceite novas conexoes de entrada (Continuando assim a escuta
da porta)
        $socketGPS = socket_accept($socket) or die("\n Nao Foi Possivel Aceitar Uma
Conexao De Entrada! - iGeoRastreador \n");

    }
} else {

    //echo "\n iGeoRastreador: Importador de dados Finalizado com Sucesso! \n";
    // Fechar o socket "filho"
    //socket_close($socketGPS);
    // parar o loop
    // break;

}

```

```
} while (true);
```

```
// fechar o socket primario
```

```
socket_close($socket);
```

```
echo "\n Fim do Importador de Dados - iGeoRastreador!! \n";
```

```
?>
```

APÊNDICE B – DISPONIBILIZAÇÃO DO CÓDIGO FONTE DO PROJETO

Para facilitar a distribuição do código fonte do projeto desenvolvido, há um repositório criado no serviço de repositórios da empresa google.

O mesmo pode ser acessado através do endereço:

<http://code.google.com/p/igeorastreador/>

Todas as atualizações e histórico de revisões podem ser acompanhadas no endereço do repositório.

O sistema está online e pode ser acessado no endereço:

<http://www.ueiner.com/igeorastreador/>

Para os professores e na versão final, um CD disponibilizado juntamente a este documento para que a instalação possa ser realizada por quem estiver interessado em analisar os códigos fontes gerados.

Um pré-requisito para a instalação do sistema é a instalação da biblioteca FrameCalixto. O documento com os passos para a instalação do framework está anexo a este documento.

APENDICE C – CÓDIGO PARA CRIAÇÃO DA BASE DE DADOS

```
--
-- PostgreSQL database dump
--

-- Started on 2010-11-21 22:11:41 BRST

SET statement_timeout = 0;
SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

SET search_path = public, pg_catalog;

--
-- TOC entry 19 (class 1255 OID 20672)
-- Dependencies: 381 6
-- Name: accent_remove(character varying); Type: FUNCTION; Schema: public; Owner:
postgres
--

CREATE FUNCTION accent_remove(text_input character varying) RETURNS character
varying
    LANGUAGE plpgsql STRICT
    AS $$

        DECLARE

            text_output varchar;

        BEGIN

            text_output = text_input;

            text_output = replace(text_output, 'Á', 'A');

            text_output = replace(text_output, 'á', 'a');

            text_output = replace(text_output, 'à', 'a');

            text_output = replace(text_output, 'À', 'A');

            text_output = replace(text_output, 'â', 'a');

            text_output = replace(text_output, 'Â', 'A');

            text_output = replace(text_output, 'ä', 'a');
```



```

text_output = replace(text_output,'Ä','A');
text_output = replace(text_output,'ä','a');
text_output = replace(text_output,'Å','A');
text_output = replace(text_output,'å','a');
text_output = replace(text_output,'Ă','A');
text_output = replace(text_output,'ö','o');
text_output = replace(text_output,'é','e');
text_output = replace(text_output,'É','E');
text_output = replace(text_output,'È','E');
text_output = replace(text_output,'è','e');
text_output = replace(text_output,'Ê','E');
text_output = replace(text_output,'ê','e');
text_output = replace(text_output,'Ë','E');
text_output = replace(text_output,'ë','e');
text_output = replace(text_output,'Í','I');
text_output = replace(text_output,'Í','I');
text_output = replace(text_output,'ì','i');
text_output = replace(text_output,'Ì','I');
text_output = replace(text_output,'î','i');
text_output = replace(text_output,'Î','I');
text_output = replace(text_output,'ï','i');
text_output = replace(text_output,'Ï','I');
text_output = replace(text_output,'ñ','n');
text_output = replace(text_output,'Ñ','N');
text_output = replace(text_output,'ó','o');

```

```

text_output = replace(text_output,'Ó','O');
text_output = replace(text_output,'Ò','O');
text_output = replace(text_output,'ò','o');
text_output = replace(text_output,'Ô','O');
text_output = replace(text_output,'ô','o');
text_output = replace(text_output,'Ö','O');
text_output = replace(text_output,'ö','o');
text_output = replace(text_output,'õ','o');
text_output = replace(text_output,'Õ','O');
text_output = replace(text_output,'Ú','U');
text_output = replace(text_output,'ú','u');
text_output = replace(text_output,'ù','u');
text_output = replace(text_output,'Û','U');
text_output = replace(text_output,'û','u');
text_output = replace(text_output,'Û','U');
text_output = replace(text_output,'ü','u');
text_output = replace(text_output,'Ü','U');
text_output = replace(text_output,'ý','y');
text_output = replace(text_output,'Ý','Y');
text_output = replace(text_output,'ÿ','y');
text_output = replace(text_output,'Ç','C');
text_output = replace(text_output,'ç','c');

return text_output;

end; $$;

```

```
ALTER FUNCTION public.accent_remove(text_input character varying) OWNER TO
postgres;
```

```
SET default_tablespace = '';
```

```
SET default_with_oids = false;
```

```
--
-- TOC entry 1568 (class 1259 OID 20673)
-- Dependencies: 6
-- Name: acesso; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE acesso (
    aces_id_acesso numeric(10,0) NOT NULL,
    aces_id_perfil numeric(10,0),
    aces_id_usuario numeric(10,0),
    aces_nm_acesso character varying(500) NOT NULL,
    aces_dt_inicio timestamp without time zone,
    aces_dt_fim timestamp without time zone
);
```

```
ALTER TABLE public.acesso OWNER TO postgres;
```

```
--
-- TOC entry 1569 (class 1259 OID 20682)
-- Dependencies: 6
-- Name: detalhe_veiculo; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE detalhe_veiculo (
    deve_id_detalhe_veiculo numeric(10,0) NOT NULL,
    deve_id_veiculo numeric(10,0) NOT NULL,
    deve_cs_tipo_detalhe character varying(2),
    deve_dt_detalhe timestamp without time zone,
    deve_vl_detalhe numeric
);
```

```
ALTER TABLE public.detalhe_veiculo OWNER TO postgres;
```

```
--
-- TOC entry 1614 (class 1259 OID 21140)
-- Dependencies: 6
-- Name: encomenda; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE encomenda (
    enco_id_encomenda numeric(10,0) NOT NULL,
```

```

enco_id_percurso numeric(10,0),
enco_id_pessoa numeric(10,0),
enco_cd_rastreamento character varying(20),
enco_ds_encomenda character varying(50),
enco_te_observacao character varying,
enco_dt_cadastro timestamp without time zone,
enco_cs_entregue numeric(1,0),
enco_dt_entrega timestamp without time zone
);

```

```

ALTER TABLE public.encomenda OWNER TO postgres;

```

```

--
-- TOC entry 1570 (class 1259 OID 20694)
-- Dependencies: 6
-- Name: estado; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

```

```

CREATE TABLE estado (
    esta_id_estado numeric(2,0) NOT NULL,
    esta_sg_sigla character varying(2) NOT NULL,
    esta_nm_estado character varying(70) NOT NULL
);

```

```

ALTER TABLE public.estado OWNER TO postgres;

```

```

--
-- TOC entry 1571 (class 1259 OID 20697)
-- Dependencies: 6
-- Name: gps_tracker; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

```

```

CREATE TABLE gps_tracker (
    gps_id_tracker integer NOT NULL,
    gps_nr_serial character varying(25),
    gps_nr_autorizado character varying(25),
    gps_cs_tipo_mensagem character varying(25),
    gps_nr_tempo_utc character varying(25),
    gps_cs_estado_sinal character varying(25),
    gps_nr_latitude character varying(25),
    gps_cs_latitude_tipo character varying(25),
    gps_cs_longitude character varying(25),
    gps_cs_longitude_tipo character varying(25),
    gps_nr_velocidade character varying(25),
    gps_cs_curso character varying(25),
    gps_dt_gps character varying(25),
    gps_vl_variacao_magnetica character varying(25),
    gps_cs_modos_gps character varying(25),

```

```

gps_nr_checksum_gprmc character varying(25),
gps_cs_indicador_gps character varying(25),
gps_nr_imei character varying(25),
gps_nr_gps character varying(25),
gps_nr_altitude character varying(25),
gps_vl_bateria character varying(25),
gps_cs_carregando character varying(25),
gps_nr_tamanho_string character varying(25),
gps_nr_checksum character varying(25),
gps_nr_codigo_pais character varying(25),
gps_nr_codigo_rede character varying(25),
gps_nr_codigo_area character varying(25),
gps_nr_id_celular character varying(25)
);

```

```

ALTER TABLE public.gps_tracker OWNER TO postgres;

```

```

--
-- TOC entry 1572 (class 1259 OID 20703)
-- Dependencies: 1571 6
-- Name: gps_tracker_gps_id_tracker_seq; Type: SEQUENCE; Schema: public; Owner:
postgres
--

```

```

CREATE SEQUENCE gps_tracker_gps_id_tracker_seq
START WITH 1
INCREMENT BY 1
NO MAXVALUE
NO MINVALUE
CACHE 1;

```

```

ALTER TABLE public.gps_tracker_gps_id_tracker_seq OWNER TO postgres;

```

```

--
-- TOC entry 1976 (class 0 OID 0)
-- Dependencies: 1572
-- Name: gps_tracker_gps_id_tracker_seq; Type: SEQUENCE OWNED BY; Schema: public;
Owner: postgres
--

```

```

ALTER          SEQUENCE          gps_tracker_gps_id_tracker_seq          OWNED          BY
gps_tracker.gps_id_tracker;

```

```

--
-- TOC entry 1977 (class 0 OID 0)
-- Dependencies: 1572

```

```
-- Name: gps_tracker_gps_id_tracker_seq; Type: SEQUENCE SET; Schema: public; Owner:
postgres
```

```
--
```

```
SELECT pg_catalog.setval('gps_tracker_gps_id_tracker_seq', 1, true);
```

```
--
```

```
-- TOC entry 1573 (class 1259 OID 20705)
```

```
-- Dependencies: 6
```

```
-- Name: local; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
```

```
--
```

```
CREATE TABLE local (
    loca_id_local numeric(10,0) NOT NULL,
    loca_id_tipo_local numeric(10,0),
    loca_nm_local character varying,
    loca_nr_latitude character varying,
    loca_nr_longitude character varying
);
```

```
ALTER TABLE public.local OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1574 (class 1259 OID 20711)
```

```
-- Dependencies: 6
```

```
-- Name: log_acesso; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
```

```
--
```

```
CREATE TABLE log_acesso (
    loac_id_log_acesso numeric(4,0) NOT NULL,
    loac_id_usuario numeric(4,0),
    loac_tx_url character varying(3000),
    loac_dt_acesso timestamp without time zone,
    loac_tx_ip character varying(25)
);
```

```
ALTER TABLE public.log_acesso OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1575 (class 1259 OID 20717)
```

```
-- Dependencies: 6
```

```
-- Name: menu; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
```

```
--
```

```
CREATE TABLE menu (
    menu_id_menu numeric(4,0) NOT NULL,
    menu_nm_menu character varying(50) NOT NULL,
```

```

    menu_tx_descricao character varying(200)
);

```

```

ALTER TABLE public.menu OWNER TO postgres;

```

```

--
-- TOC entry 1576 (class 1259 OID 20720)
-- Dependencies: 6
-- Name: menu_item; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

```

```

CREATE TABLE menu_item (
    meit_id_menu_item numeric(4,0) NOT NULL,
    meit_id_pai numeric(4,0) NOT NULL,
    meit_id_menu numeric(4,0) NOT NULL,
    meit_nr_posicao numeric(10,0),
    meit_tx_url character varying(1024),
    meit_nm_alvo character varying(20),
    meit_tx_imagem character varying(1024),
    meit_bo_destravado character varying(3),
    meit_nm_menu_item character varying(50),
    meit_tx_descricao character varying(3000)
);

```

```

ALTER TABLE public.menu_item OWNER TO postgres;

```

```

--
-- TOC entry 1577 (class 1259 OID 20726)
-- Dependencies: 6
-- Name: ocorrencia; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

```

```

CREATE TABLE ocorrencia (
    ocor_id_ocorrencia numeric(10,0) NOT NULL,
    ocor_id_veiculo numeric(10,0) NOT NULL,
    ocor_id_tipo_ocorrencia numeric(10,0),
    ocor_dt_ocorrencia timestamp without time zone,
    ocor_te_observacao character varying,
    ocor_bo_lancamento_automatico numeric(1,0)
);

```

```

ALTER TABLE public.ocorrencia OWNER TO postgres;

```

```

--
-- TOC entry 1578 (class 1259 OID 20732)
-- Dependencies: 6
-- Name: percurso; Type: TABLE; Schema: public; Owner: postgres; Tablespace:

```

--

```
CREATE TABLE percurso (
    perc_id_percurso numeric(10,0) NOT NULL,
    perc_nm_percurso character varying(50) NOT NULL,
    perc_id_local_inicio numeric(10,0),
    perc_id_local_fim numeric(10,0),
    perc_cs_tipo_percurso character varying(2)
);
```

```
ALTER TABLE public.percurso OWNER TO postgres;
```

--

```
-- TOC entry 1579 (class 1259 OID 20735)
-- Dependencies: 6
-- Name: perfil; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE perfil (
    perf_id_perfil numeric(10,0) NOT NULL,
    perf_nm_perfil character varying(60) NOT NULL,
    perf_bo_log_acesso character varying(3)
);
```

```
ALTER TABLE public.perfil OWNER TO postgres;
```

--

```
-- TOC entry 1580 (class 1259 OID 20738)
-- Dependencies: 6
-- Name: pessoa; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE pessoa (
    pess_id_pessoa numeric(10,0) NOT NULL,
    pess_cs_pessoa character varying(2) NOT NULL,
    pess_nm_pessoa character varying(100) NOT NULL,
    pess_nr_documento character varying(25),
    pess_nr_cep character varying(8),
    pess_nr_telefone character varying(21),
    pess_nr_telefone2 character varying(21),
    pess_nr_telefone3 character varying(21),
    pess_id_estado numeric(2,0),
    pess_nm_municipio character varying(100),
    pess_nm_bairro character varying(100),
    pess_tx_endereco character varying(254),
    pess_nm_email character varying(255),
    pess_nm_site character varying(1000)
);
```



```
ALTER TABLE public.pessoa OWNER TO postgres;
```

```
--
-- TOC entry 1612 (class 1259 OID 20997)
-- Dependencies: 6
-- Name: rastreamento; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--
```

```
CREATE TABLE rastreamento (
    rast_id_rastreamento numeric(10,0) NOT NULL,
    rast_id_veiculo numeric(10,0) NOT NULL,
    rast_id_encomenda numeric(10,0),
    rast_dt_rastreamento timestamp without time zone,
    rast_hr_rastreamento character varying(10)
);
```

```
ALTER TABLE public.rastreamento OWNER TO postgres;
```

```
--
-- TOC entry 1581 (class 1259 OID 20744)
-- Dependencies: 6
-- Name: sq_; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_
    START WITH 1
    INCREMENT BY 1
    NO MAXVALUE
    NO MINVALUE
    CACHE 1;
```

```
ALTER TABLE public.sq_ OWNER TO postgres;
```

```
--
-- TOC entry 1978 (class 0 OID 0)
-- Dependencies: 1581
-- Name: sq_; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_', 1, false);
```

```
--
-- TOC entry 1582 (class 1259 OID 20746)
-- Dependencies: 6
-- Name: sq_acesso; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_acesso
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_acesso OWNER TO postgres;
```

--

```
-- TOC entry 1979 (class 0 OID 0)
-- Dependencies: 1582
-- Name: sq_acesso; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_acesso', 462, true);
```

--

```
-- TOC entry 1583 (class 1259 OID 20748)
-- Dependencies: 6
-- Name: sq_cliente; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_cliente
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_cliente OWNER TO postgres;
```

--

```
-- TOC entry 1980 (class 0 OID 0)
-- Dependencies: 1583
-- Name: sq_cliente; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_cliente', 1, false);
```

--

```
-- TOC entry 1584 (class 1259 OID 20750)
-- Dependencies: 6
-- Name: sq_detalhe_veiculo; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_detalhe_veiculo
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_detalhe_veiculo OWNER TO postgres;
```

--

```
-- TOC entry 1981 (class 0 OID 0)
-- Dependencies: 1584
-- Name: sq_detalhe_veiculo; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_detalhe_veiculo', 1, false);
```

--

```
-- TOC entry 1613 (class 1259 OID 21138)
-- Dependencies: 6
-- Name: sq_encomenda; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_encomenda
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_encomenda OWNER TO postgres;
```

--

```
-- TOC entry 1982 (class 0 OID 0)
-- Dependencies: 1613
-- Name: sq_encomenda; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_encomenda', 1, false);
```

--

```
-- TOC entry 1585 (class 1259 OID 20754)
-- Dependencies: 6
-- Name: sq_estado; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_estado
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_estado OWNER TO postgres;
```

--

```
-- TOC entry 1983 (class 0 OID 0)
-- Dependencies: 1585
-- Name: sq_estado; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_estado', 27, true);
```

--

```
-- TOC entry 1586 (class 1259 OID 20756)
-- Dependencies: 6
-- Name: sq_gps_tracker; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_gps_tracker
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_gps_tracker OWNER TO postgres;
```

--

```
-- TOC entry 1984 (class 0 OID 0)
-- Dependencies: 1586
-- Name: sq_gps_tracker; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_gps_tracker', 1, false);
```

--

```
-- TOC entry 1587 (class 1259 OID 20758)
-- Dependencies: 6
-- Name: sq_local; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_local
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_local OWNER TO postgres;
```

--

```
-- TOC entry 1985 (class 0 OID 0)
-- Dependencies: 1587
-- Name: sq_local; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_local', 1, false);
```

--

```
-- TOC entry 1588 (class 1259 OID 20760)
-- Dependencies: 6
-- Name: sq_log_acesso; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_log_acesso
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_log_acesso OWNER TO postgres;
```

--

```
-- TOC entry 1986 (class 0 OID 0)
-- Dependencies: 1588
-- Name: sq_log_acesso; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_log_acesso', 659, true);
```

--

```
-- TOC entry 1589 (class 1259 OID 20762)
-- Dependencies: 6
-- Name: sq_menu; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE SEQUENCE sq_menu
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_menu OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1987 (class 0 OID 0)
-- Dependencies: 1589
-- Name: sq_menu; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_menu', 1, false);
```

```
--
```

```
-- TOC entry 1590 (class 1259 OID 20764)
-- Dependencies: 6
-- Name: sq_menu_item; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_menu_item
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_menu_item OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1988 (class 0 OID 0)
-- Dependencies: 1590
-- Name: sq_menu_item; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_menu_item', 1, false);
```

```
--
```

```
-- TOC entry 1591 (class 1259 OID 20766)
-- Dependencies: 6
-- Name: sq_ocorrenzia; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_ocorrencia
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_ocorrencia OWNER TO postgres;
```

--

```
-- TOC entry 1989 (class 0 OID 0)
-- Dependencies: 1591
-- Name: sq_ocorrencia; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_ocorrencia', 1, false);
```

--

```
-- TOC entry 1592 (class 1259 OID 20768)
-- Dependencies: 6
-- Name: sq_percurso; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_percurso
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_percurso OWNER TO postgres;
```

--

```
-- TOC entry 1990 (class 0 OID 0)
-- Dependencies: 1592
-- Name: sq_percurso; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_percurso', 1, false);
```

--

```
-- TOC entry 1593 (class 1259 OID 20770)
-- Dependencies: 6
-- Name: sq_perfil; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_perfil
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_perfil OWNER TO postgres;
```

--

```
-- TOC entry 1991 (class 0 OID 0)
-- Dependencies: 1593
-- Name: sq_perfil; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_perfil', 2, true);
```

--

```
-- TOC entry 1594 (class 1259 OID 20772)
-- Dependencies: 6
-- Name: sq_pessoa; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_pessoa
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_pessoa OWNER TO postgres;
```

--

```
-- TOC entry 1992 (class 0 OID 0)
-- Dependencies: 1594
-- Name: sq_pessoa; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_pessoa', 6, true);
```

--

```
-- TOC entry 1611 (class 1259 OID 20995)
-- Dependencies: 6
-- Name: sq_rastreamento; Type: SEQUENCE; Schema: public; Owner: postgres
```



```
--
```

```
CREATE SEQUENCE sq_rastreamento
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_rastreamento OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1993 (class 0 OID 0)
-- Dependencies: 1611
-- Name: sq_rastreamento; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_rastreamento', 1, false);
```

```
--
```

```
-- TOC entry 1595 (class 1259 OID 20774)
-- Dependencies: 6
-- Name: sq_rota; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_rota
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_rota OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1994 (class 0 OID 0)
-- Dependencies: 1595
-- Name: sq_rota; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_rota', 1, false);
```

```
--
```

```
-- TOC entry 1596 (class 1259 OID 20776)
-- Dependencies: 6
-- Name: sq_tipo_local; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_tipo_local
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_tipo_local OWNER TO postgres;
```

--

```
-- TOC entry 1995 (class 0 OID 0)
-- Dependencies: 1596
-- Name: sq_tipo_local; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_tipo_local', 1, true);
```

--

```
-- TOC entry 1597 (class 1259 OID 20778)
-- Dependencies: 6
-- Name: sq_tipo_ocorrendia; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_tipo_ocorrendia
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_tipo_ocorrendia OWNER TO postgres;
```

--

```
-- TOC entry 1996 (class 0 OID 0)
-- Dependencies: 1597
-- Name: sq_tipo_ocorrendia; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_tipo_ocorrendia', 1, false);
```

--

```
-- TOC entry 1598 (class 1259 OID 20780)
-- Dependencies: 6
-- Name: sq_trecho; Type: SEQUENCE; Schema: public; Owner: postgres
```

```
--
```

```
CREATE SEQUENCE sq_trecho
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_trecho OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1997 (class 0 OID 0)
-- Dependencies: 1598
-- Name: sq_trecho; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_trecho', 1, false);
```

```
--
```

```
-- TOC entry 1609 (class 1259 OID 20980)
-- Dependencies: 6
-- Name: sq_trecho_rastreamento; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_trecho_rastreamento
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_trecho_rastreamento OWNER TO postgres;
```

```
--
```

```
-- TOC entry 1998 (class 0 OID 0)
-- Dependencies: 1609
-- Name: sq_trecho_rastreamento; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_trecho_rastreamento', 1, false);
```

```
--
```

```
-- TOC entry 1599 (class 1259 OID 20782)
-- Dependencies: 6
-- Name: sq_usuario; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_usuario
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_usuario OWNER TO postgres;
```

--

```
-- TOC entry 1999 (class 0 OID 0)
-- Dependencies: 1599
-- Name: sq_usuario; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_usuario', 2, true);
```

--

```
-- TOC entry 1600 (class 1259 OID 20784)
-- Dependencies: 6
-- Name: sq_usuario_perfil; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_usuario_perfil
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_usuario_perfil OWNER TO postgres;
```

--

```
-- TOC entry 2000 (class 0 OID 0)
-- Dependencies: 1600
-- Name: sq_usuario_perfil; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_usuario_perfil', 2, true);
```

--

```
-- TOC entry 1601 (class 1259 OID 20786)
-- Dependencies: 6
-- Name: sq_veiculo; Type: SEQUENCE; Schema: public; Owner: postgres
```

--

```
CREATE SEQUENCE sq_veiculo
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_veiculo OWNER TO postgres;
```

--

```
-- TOC entry 2001 (class 0 OID 0)
-- Dependencies: 1601
-- Name: sq_veiculo; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_veiculo', 1, true);
```

--

```
-- TOC entry 1602 (class 1259 OID 20788)
-- Dependencies: 6
-- Name: sq_veiculos; Type: SEQUENCE; Schema: public; Owner: postgres
--
```

```
CREATE SEQUENCE sq_veiculos
  START WITH 1
  INCREMENT BY 1
  NO MAXVALUE
  NO MINVALUE
  CACHE 1;
```

```
ALTER TABLE public.sq_veiculos OWNER TO postgres;
```

--

```
-- TOC entry 2002 (class 0 OID 0)
-- Dependencies: 1602
-- Name: sq_veiculos; Type: SEQUENCE SET; Schema: public; Owner: postgres
--
```

```
SELECT pg_catalog.setval('sq_veiculos', 1, false);
```

--

```
-- TOC entry 1603 (class 1259 OID 20790)
-- Dependencies: 6
-- Name: tipo_local; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
```

--

```
CREATE TABLE tipo_local (
    tilo_id_tipo_local numeric(10,0) NOT NULL,
    tilo_ds_tipo_local character varying(60)
);
```

```
ALTER TABLE public.tipo_local OWNER TO postgres;
```

--

-- TOC entry 1604 (class 1259 OID 20793)

-- Dependencies: 6

-- Name: tipo_ocorrencia; Type: TABLE; Schema: public; Owner: postgres; Tablespace:

--

```
CREATE TABLE tipo_ocorrencia (
    tioc_id_tipo_ocorrencia numeric(10,0) NOT NULL,
    tioc_ds_tipo_ocorrencia character varying(60)
);
```

```
ALTER TABLE public.tipo_ocorrencia OWNER TO postgres;
```

--

-- TOC entry 1605 (class 1259 OID 20796)

-- Dependencies: 6

-- Name: trecho; Type: TABLE; Schema: public; Owner: postgres; Tablespace:

--

```
CREATE TABLE trecho (
    trec_id_trecho numeric(10,0) NOT NULL,
    trec_id_percurso numeric(10,0),
    trec_nr_latitude_inicial character varying(20),
    trec_nr_longitude_inicial character varying(20),
    trec_nr_latitude_final character varying(20),
    trec_nr_longitude_final character varying(20),
    trec_nr_ordem numeric(2,0)
);
```

```
ALTER TABLE public.trecho OWNER TO postgres;
```

--

-- TOC entry 1610 (class 1259 OID 20982)

-- Dependencies: 6

-- Name: trecho_rastreamento; Type: TABLE; Schema: public; Owner: postgres; Tablespace:

--

```
CREATE TABLE trecho_rastreamento (
```

```

    trra_id_trecho numeric(10,0) NOT NULL,
    trra_id_rastreamento numeric(10,0),
    trra_nr_latitude character varying,
    trra_nr_longitude character varying,
    trra_nr_altitude character varying,
    trra_nr_velocidade character varying,
    trra_nr_ordem numeric(10,0)
);

```

```

ALTER TABLE public.trecho_rastreamento OWNER TO postgres;

```

```

--
-- TOC entry 1606 (class 1259 OID 20799)
-- Dependencies: 6
-- Name: usuario; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

```

```

CREATE TABLE usuario (
    usua_id_usuario numeric(10,0) NOT NULL,
    usua_id_pessoa numeric(10,0) NOT NULL,
    usua_nm_login character varying(25) NOT NULL,
    usua_nm_senha character varying(32) NOT NULL
);

```

```

ALTER TABLE public.usuario OWNER TO postgres;

```

```

--
-- TOC entry 1607 (class 1259 OID 20802)
-- Dependencies: 6
-- Name: usuario_perfil; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

```

```

CREATE TABLE usuario_perfil (
    usup_id_usuario_perfil numeric(10,0) NOT NULL,
    usup_id_usuario numeric(10,0) NOT NULL,
    usup_id_perfil numeric(10,0) NOT NULL
);

```

```

ALTER TABLE public.usuario_perfil OWNER TO postgres;

```

```

--
-- TOC entry 1608 (class 1259 OID 20805)
-- Dependencies: 6
-- Name: veiculo; Type: TABLE; Schema: public; Owner: postgres; Tablespace:
--

```

```

CREATE TABLE veiculo (

```

```
veic_id_veiculo numeric(10,0) NOT NULL,  
veic_id_pessoa numeric(10,0),  
veic_dt_inicio_operacao timestamp without time zone,  
veic_nr_imei numeric(15,0),  
veic_nr_placa character varying(8),  
veic_te_descricao character varying(200)  
);
```

```
ALTER TABLE public.veiculo OWNER TO postgres;
```

```
--  
-- TOC entry 1892 (class 2604 OID 20808)  
-- Dependencies: 1572 1571  
-- Name: gps_id_tracker; Type: DEFAULT; Schema: public; Owner: postgres  
--
```

```
ALTER TABLE gps_tracker ALTER COLUMN gps_id_tracker SET DEFAULT  
nextval('gps_tracker_gps_id_tracker_seq'::regclass);
```


ANEXO I – MANUAL DE INSTALAÇÃO DO FRAMECALIXTO

Como sempre em informática, quando se trata de software sempre ouvimos estas duas palavras “instalação” que consiste na alocação do software dentro da máquina e a “configuração” que é a forma de ensinar ao software como ele deverá trabalhar.

Venho trabalhando desde o início para que o FrameCalixto seja fácil de instalar e configurar, espero que esta seja uma etapa que os desenvolvedores não fiquem perdendo tempo para começar a “botar a mão na massa”.

Este post tem o objetivo de explicar como deve ser a instalação e a configuração do FrameCalixto.

- * Faça o download do arquivo compactado na página principal do projeto, ou baixe via svn.
 1. Linux svn checkout <http://framecalixto.googlecode.com/svn/trunk/> framecalixto-read-only
 2. Windows utilize o Tortoise SVN
- * Descompacte em uma pasta do apache que execute PHP.
- * Para a primeira execução determine as configurações iniciais do sistema e do banco de dados.
- * Execute a recriação do banco de dados.
- * Cadastre uma pessoa.
- * Cadastre um usuário.
- * Libere os acessos do usuário.
- * Retorne a tela de configurações do sistema e trave o controle de acesso.

Basicamente este são os passos para a utilização do FrameCalixto.

Escrito Por: Calixto Jorge.